

Tietokonekulttuurin erikoislehti



Koodaa demo
GAME BOYLL

**MIKKO
HYPPÖNEN**
viikko jolloin
WannaCry iski

Verkon suojeleija
**DONNIE
WERNER**

- 3 Pääkirjoitus**
- 4 Donnie Werner – hakkerista tietoturvakonsultiksi**
Entinen mustahattu kertoo, mikä nykykoodareissa on vialla.
- 12 Kolumni: Mikko Hyppönen**
Viikko tietoturvasekatyömiehen elämästä.
- 13 RuuviTag**
Kotimaista älyä esineverkkoon.
- 14 Nokia 3310**
Tuhoutumaton ihmease vai muistojen kultaama tusinaluuri? Jatkoa pukkua.
- 16 Ystävämme Petri**
Petri-verkot takaavat muun muassa lentokoneiden ja ydinvoimaloiden toimintaa.
- 19 Kolumni: Janne Sirén**
Standardisodan taide.
- 20 Kuvat kuntoon**
Valokuvien jälkikäsitteily on hyvä taito myös harrastelijan työkalupakissa.
- 25 Keinotodellisuuspäivitys ja SD-512**
Uusi Lyhyet-palstamme saa jatkoa.
- 26 Vattukoneella lentoja seuraamaan**
Laita Raspberry Pi tutkimaan ilmatilaa.
- 28 NodeMCU – avain esineverkkoon**
Oma IoT-laite syntyy helpommin kuin luulitkaan.
- 34 Muistatko, kun monitori säteili?**
Joskus kuvaputkien turvallisuudesta kiersi kummallisia käsityksiä.
- 36 Stringologia**
Bioinformatiikan tutkimuksen taustatiedot tutuiksi.
- 40 Surkeat suomipelit**
JOKin pelijutussa suomipelien synnintunnustus.
- 44 Ilmaisia lautapelejä verkossa – mutta mihin hintaan?**
Lautapeliin siirtäminen tietokoneelle luo uusia mahdollisuuksia ja ongelmia.
- 48 Sims**
Hittipelin pinnan alta paljastuu eloisa ja erikoinen yhteisö.
- 51 Sarjakuva**
Turrikaanien yö tekee kirjallisuustutkimusta.
- 52 Wolfensteinin perintö**
Kertomus pienestä ohjelmistotalosta, joka löysi kultaa.
- 56 Kolumni: Mikko Heinonen**
Ihana, kamala Amiga.
- 57 Lisänostetta haukan liitton – CT60e**
Myös Atarin saa turboahdettua.
- 60 Oi muistatko vielä sen SIMMin?**
Miten selvitetään tuntemattoman muistikamman koko?
- 62 Ei näin!**
Acorn saavutti paljon mutta myös hukkasi mahdollisuuksia.
- 64 TAUON paikka!**
Poista itsesi koneelta – koneellisesti.
- 65 Koodia Game Boylelle kotikonstein**
Taskupelikonsoli taipuu myös demontekoon.
- 69 Peliklassikko BASICilla: Star Trek**
Vanhasta pelistä voi edelleen oppia ohjelmointia.
- 72 JOKstoriaa**
Pelikonkarin muistelmapäälta alkaa.
- 74 Skrolli Party 2017**
Pizzaa ja demoja – niistä on hyvät juhlat tehty.



Tapio Berschewsky
päätoimittaja

Onnea ja kiitollisuutta

Skrollin ensimmäinen kevät uudella luotsauksella on edennyt leppoisasti. Siitä olemme ensisijaisesti kiitollisia avustajillemme, jotka puurtavat väsymättömästi hyvien artikkelien eteen. Toimitus huhkii lehden kasaan, mutta ilman avustajia sitä ei olisi.

Kevään mittaan järjestettiin siksi myös ensimmäinen Skrolli Party – uusi demotapahtuma, joka on suunnattu erityisesti lehden avustajille. Tapahtuma järjestettiin avustajien aloitteesta ja heitä varten kiitoksena huikeasta työstä, jota avustajat tekevät Skrollin eteen.

Partyjen ideoisija ja isäntä sekä pitkäaikainen skrolliitti **Mikko Rasa** antoi juhlien käyttöön oman kotinsa. Juhlissa järjestetty digitaalisen taiteen kompo, musiikkiesitykset sekä demoshow pidettiin Mikon alakerran elokuvateatterissa. Kuvia ja tunnelmia bileistä löydät lehden viimeiseltä aukeamalta.

Tietää olevansa onnekas, kun saa tehdä hommia näin mukavan porukan kanssa, joka haluaa yhteisestä palosta pitää tietokonekulttuurin lippua korkealla – ja jaksaa vielä juhliakin yhdessä.

Myös Skrollin läsnäolo alkuvuodesta hakkeritapahtuma Disobeyssa poiki paljon hyvää. Paitsi että syvensimme suhdettamme organisaatioon, jonka tavoitteet ovat hyvin linjassa omiemme kanssa, satuimme jälkibileissä samaan pöytään **Donnie Wernerin** kanssa ja saimme sovittua huikean haastatteluilaisuuden vanhan kaartin hakkerin kanssa.

Disobeyn yhteydessä keksimme myös kömpelön "esineiden internet" -termin ja uuden kivan sanan vastenmielisen IoT-lyhennyksen tilalle: esineverkko. Se rullaa kieleltä mukavasti. Kekkauksestamme tykättiin somessa Aamulehden päätoimittajaa myöten.

Ja miten edes voi käydä sellainen tuuri, että pari kuukautta aiemmin pyytää **Mikko Hyppöstä** toimittamaan kolumnin juuri sillä viikolla, jolla WannaCry sattumalta tapahtuu. 🐛

Skrolli

Tietokonekulttuurin erikoislehti

Yhteydenotot toimitus@skrolli.fi
Ircnet: #skrolli
skrolli.fi

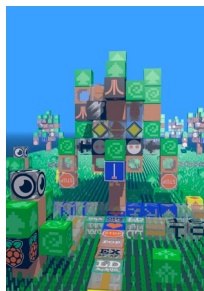
Vastaava päätoimittaja Mikko Heinonen
Päätoimittaja Tapio Berschewsky
Toimituspäällikkö Valhe Kouneli
Taiteellinen johtaja, taitto Nasu Viljanmaa
Digipäälliköt Toni Kuokkanen
Janne Sirén
Taitto Manu Pärssinen
Kuvatoimittaja Laura Pesola
Mediamyynti Jari Jaanto
Talous Anssi Kolehmainen

Muu toimitus Jarno Niklas Alanko, Ville-Matias Heikkilä, Jukka O. Kauppinen, Ronja Koistinen, Ninnu Koskenalho, Sakari Lönn, Suvu Sivulainen.

Tämän numeron avustajat Mette Erikilä, Mikael Heikkanen, Mikko Hyppönen, Ville Jouppi, Emma Kantanen, Toni Kortelahti, Markus Kuula, Pinja Kuusela, Jarkko Lehti, Jarno Lehtinen, Miikka Lehtonen, Sakari Leppä, Vesa Linja-aho, Mitol Meerna, Milsa, Jarkko Nääs, Susanna Rantanen, Mikko Rasa, Kimmo Rinta-Pollari, Mikko Torvinen, Ismo Utriainen, Joni Valtanen, Samuli Vuorinen, Antti Ylikoski, Mika Yrjölä

Julkaisija Skrolli ry

Painopaikka Hämeen Kirjapaino, Tampere,
ISSN 2323-8992 (painettu)
ISSN 2323-900X (verkkójulkaisu)



4041 0209
Painotuote

HÄMEEN KIRJAPAINO OY

Etukannen kuva:
Ville-Matias Heikkilä



Donnie Werner

hakkerista tietoturvakonsultiksi

”Jo pelkästään sillä, mitä minä ja ystäväni tiedämme, voisi laittaa maailman polvilleen.”

Teksti: Valhe Kouneli (lainaukset kääntänyt englannista suomeksi Mikko Heinonen)

Kuvat: Mette Erikkilä, Valhe Kouneli, Nasu Viljanmaa



Istumme Amsterdam-ravintolassa Helsingin Ruoholahdessa **Donnie Wernerin** kanssa. Tupakanharmaa, reilu viisikymppinen mies mustassa nahkatakissaan puhuu ”fire sale”-hyökkäyksestä. Se on mediatermi, jolla tarkoitetaan maailman infrastruktuurin kaatamista – sähkönjakelun, viestinnän ja niin edelleen.

Viimeksi Yhdysvalloissa käydessään Donnie kertoo pohtineensa hakkerikaverinsa kanssa, kuinka pitkään heiltä meni toteuttaa sellainen.

– Arvioimme, että kaksi viikkoa tiedustelua ja sitten pari päivää ja se olisi siinä. Saisimme aikaan paljon vahinkoa.

Useimpien muiden sanomana se kuulostaisi yli-itsevarmalta, mutta me haastattelijat tunnemme jo hieman miehen taustaa. Sen perusteella emme koe tarpeelliseksi kyseenalaistaa hänen kertomaansa.

Donnie, joka siis haluaa tulla kutsutuksi etunimellään, on vanhan koulukunnan hakkeri.

– Järjestelmissä on muutamia kriittisiä kohtia, joihin iskemällä ne kaatuvat. Mutta en ole pahantahtoinen.

Kuka on Donnie Werner?

Donnie tekee nykyisin penetraatio-testausta suurelle viestintäyritykselle ja suojelee kaikille tärkeitä globaaleja järjestelmiä. Mutta aikanaan hän oli mustahattuhakkeri, joka tunkeutui järjestelmiin eri puolilla maailmaa.

Nuoruudessa hänen vapaa-aikansa kului ehkä hieman epätyypillisellä tavalla. Hän kertoo lyöttäytyneensä viikonloppuisin yhteen kaverinsa kanssa

ja valinneensa tietyn maan ja sen hallituksen sivuston.

– Vähän siihen tyyliin, että ”mitäs tänään tehtäisiin”. Meillä oli usein ihan onnistuneita kokemuksia. Emme kuitenkaan rikkoneet mitään. Emme olleet poliittisesti motivoituneita.

Hakkeroinnin takana oli enemmänkin tarve koetella omia taitoja.

Verkossa Donnie tunnetaan paremmin aliaksellaan `morning_wood`. Hän on myös päätenyt Defconissa Hacker Jeopardyn triviaikysymykseen ”Kuka antoi nimen Whoppixille?”

– Whoppix oli BackTrack Linuxin edeltäjä. Nimi tulee sanoista Whitehat Knoppix.

Mies nimen takana

Tunnustus on ansaittua, sillä Donnie on vaikuttanut laajalti hakkerikulttuuriin.

Kun BackTrackia edeltävät Remote Auditor- ja Slax-projektit yhdistyivät, Donnie päätyi mukaan tekijöiden porukkaan. Hän kiittää menestystään hakkerina siitä, että sai jo nuorena muiden hakkereiden huomion ja tapasi oikeita ihmisiä.

Whoppixin lisäksi Donnie kertoo keksineensä myös Whaxin eli Whitehat Slaxin nimen.

– Muts nimesi BackTrackin, mutta kävimme siitä pitkän keskustelun. Olin projektissa mukana melkein Kalin julkaisuun asti.

Mutsillä Donnie viittaa **Mati Aharoniin** – hakkeriin, joka perusti Offensive-Securityn, johtavan hyök-

kävää turvallisuuskoulutusta tarjoavan yrityksen.

Onnettomuus

Tarjoamme Donnielle juotavaa lehden piikkiin ja tiedustelemme lisää hänen taustoistaan. Donnie palaa kertomuksessaan lapsuuteensa Kaliforniaan.

– Kasvoin käytännössä lentokentällä ja tein sen alan töitä. Useimmiten tankkasin lentokoneita. Isälläni oli huoltoyritys, ja ajattelin ryhtyä itsekin sille alalle. Jouduin kuitenkin dramaattiseen moottoripyöräonnettomuuteen vuonna 1987, ja se muutti kaikki suunnitelmat.

Hän viittaa käteensä, joka on mustan nahkatakian hihan peitossa ja johon ei tulisi muuten kiinnittäneeksi huomiota.

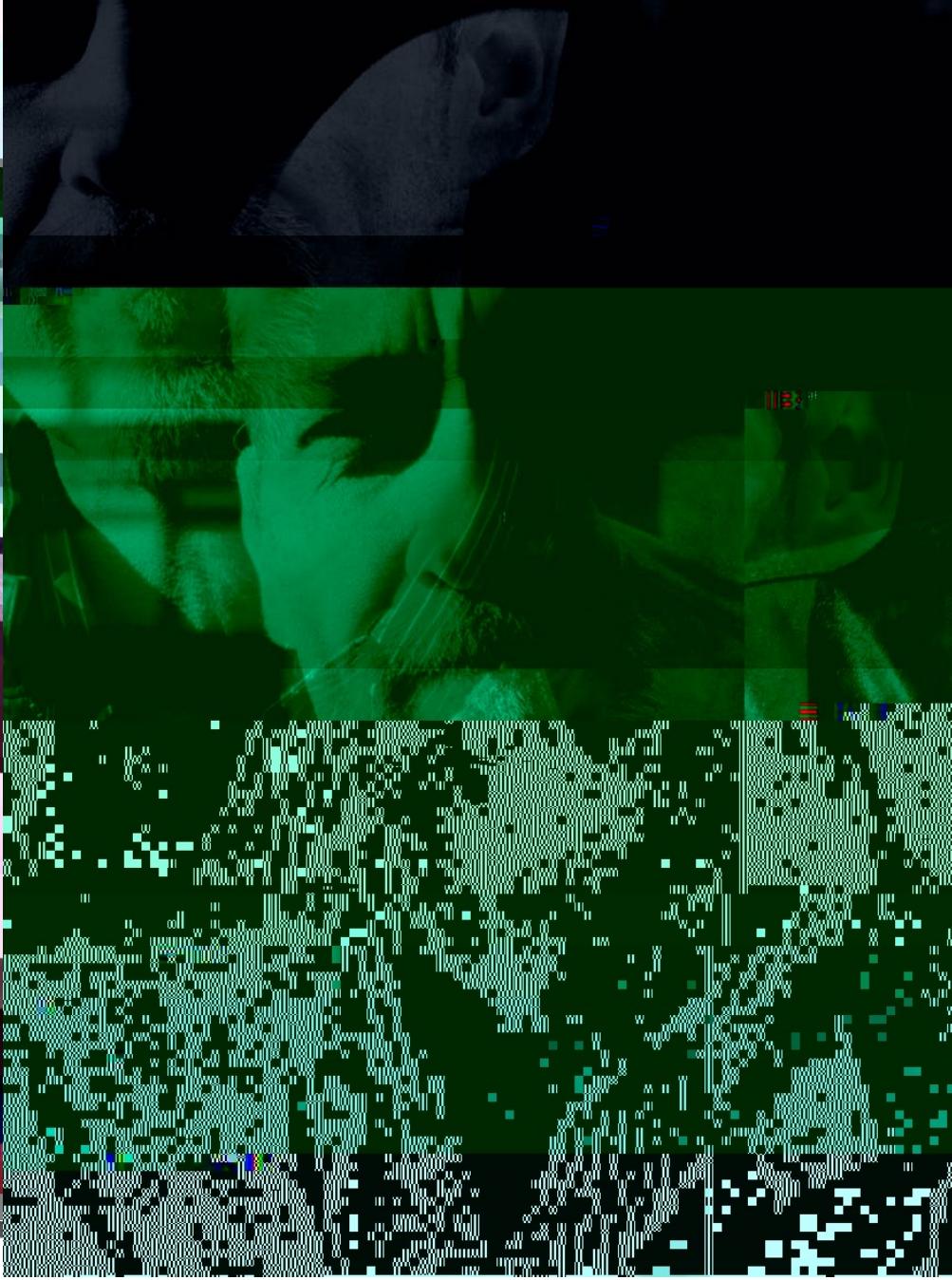
– Menetin toisen käteni hallinnan, enkä pystynyt enää tekemään fyysisiä töitä, kuten kiinteistöhuoltoa. Oli keksittävä jotain muuta.

– Olin aina se pikkupoika, joka hajottaa kaikki laitteet osiin nähdäkseen, miten ne toimivat. Vanhempana korjasin aina tietokoneita ja muuta tekniikkaa. Ura tietokoneiden parissa tuntui hyvältä ajatukselta.

Donnie kuvaili, kuinka työ tietokonekorjaamossa johti huoltopäälliköksi, ja muutamien vuoden ajan hänen päivänsä kuuluivat korjausten ja diagnostiikan tekemisen parissa. Sinä aikana ne asiat, jotka hän myöhemmin ymmärsi hakkeroinniksi ja tietoturvaksi, alkoivat tulla tutuiksi.

”Olin aina se pikkupoika, joka hajottaa kaikki laitteet osiin nähdäkseen, miten ne toimivat.”

Donnie kuvaili, kuinka työ tietokonekorjaamossa johti huoltopäälliköksi, ja muutamien vuoden ajan hänen päivänsä kuuluivat korjausten ja diagnostiikan tekemisen parissa. Sinä aikana ne asiat, jotka hän myöhemmin ymmärsi hakkeroinniksi ja tietoturvaksi, alkoivat tulla tutuiksi.



Hän myös kertoo opiskelleensa alun perin mekaanista suunnittelua, mutta jättäneensä opinnot parin vuoden jälkeen kesken. Hän kuvaillee opintojen kuitenkin suunnan- neen ajattelua ja parantaneen käsi- tystä omasta itsestä.

Varhaiset kokeilut

Vaikka Donnie ei heti tajunnutkaan kykyjään hakkerina, hän oli tehnyt sen suuntaisia asioita jo koulussa.

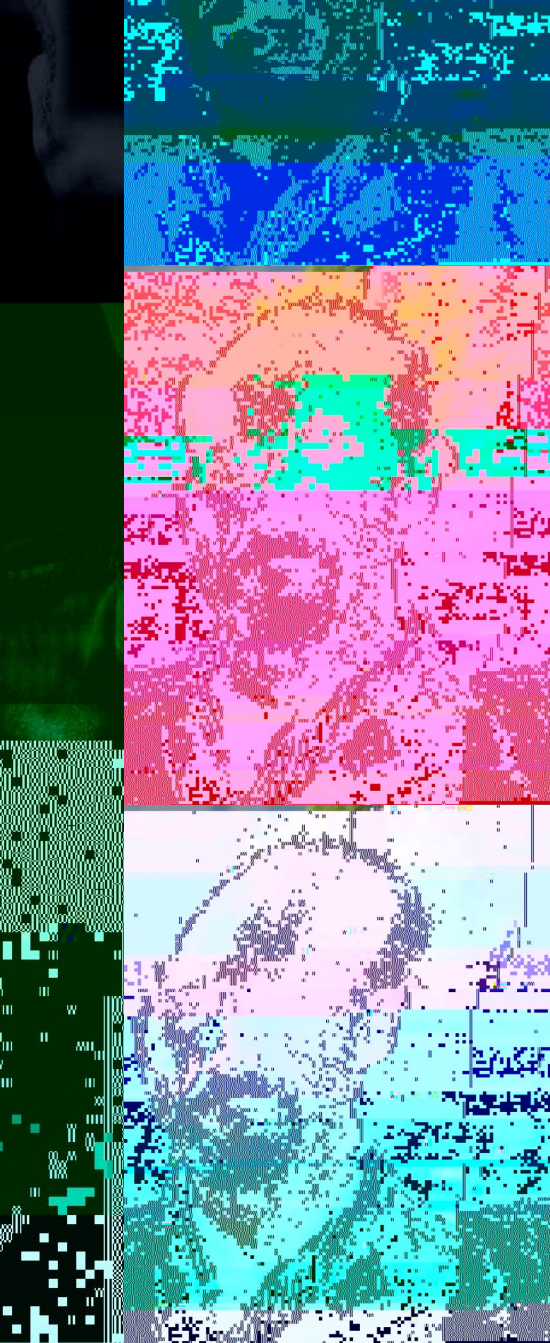
- Hakkeroin jo kun käytin tietokonetta ihan ensimmäistä kertaa.
- Olin tietokoneen alkeiskurssilla op- pimassa vuokaavioista, enkä ihan ymmärtänyt goto-silmukkaa. En vain tajunnut sitä rinnakkaisprosessointia, hän kertoo kuljettaen sormeaan kuvitteellista ohjelmakoodia pitkin selityk- sensä mukana:
- Aivoni olivat ihan jumissa.
- Luokassa oli Donnien mukaan päät-

teitä, joihin piti kirjautua palvelimen nimellä ja käyttäjätunnuksella. Hän huomasi muitakin nimiä, jotka vaikuttivat palvelimien nimiltä, eikä voinut vastustaa kiusausta. Lopulta hän pääsi sisään koulun järjestelmään.

Eivätkä hakkerointikokeilut tieten- kään jääneet siihen. Pian järjestelmiin tunkeutumisesta tuli päivittäinen har- rastus.

– Pääsin sisään modeemipuuhiin ja aloin tehdä *war dialingia*. Siinä käytetään järjestelmää, joka soittaa au- tomaattisesti eri puhelinnumeroihin. Saadessaan yhteyden tietokoneeseen se tallentaa viestit, kättelyn sekä kir- jautumiskehotteen.

– Menin töihin ja jätin sen pyörimään koko päiväksi. Illalla kotona tutkin tu- loksia ja kokeilin yhdistää ja kirjautua löytämiini järjestelmiin. Aina ei tarvin- nut edes kirjautua. Pääsin näkemään monia erilaisia käyttöjärjestelmiä.



Hänen silmänsä alkavat kiillua innostuneesti varhaisten hakkerointimuistojen kertaamisesta. Intohimoinen suhtautuminen uusien asioiden tutkiskeluun – vaikka ne olisivat kieltoimerkin takana – voi aistia koko hänen olemuksestaan.

Uteliaisuus sai hakkerin kaivautumaan yhä syvemmälle – myös hyvin kirjaimellisesti. Varojen puutteessa hän alkoi dyykata tietokonekauppojen roskiksia ja kertoo koonneensa ensimmäiset tietokoneensa dyykatuista osista.

Roskiksista löytyi laitteita, ohjelmistoja sekä käyttäjätunnuksia ja salasanoja – niitä tosin useimmiten suoraan paperille kirjoitettuna ja tulostettuina. Hän löysi myös käyttöjärjestelmiä koskevia kirjoja.

– Sattumalta olin juuri tutkimassa uutta järjestelmää ja vaikka HP-UX:n tai SCO:n kirjautumiskehotetta, ja sitten

viikkoa myöhemmin löysin roskiksesta pinon SCO-kirjoja!

Kirja nimeltä *Hacking Exposed* muutti ehkä hänen elämänsä. Siihen tutustuessaan hän oli jo saanut laajakaistayhteyden.

– Se oli minusta tosi mielenkiintoinen. Siinä kerrottiin HTML:stä ja internetin toiminnasta, mutta osa sen sisällöstä oli vähän kuin sisäpiiritietoa. Luin sen huolella ja aloin kokeilla sen tekniikoita.

– Opin paljon tietoturvatyössä käytettäviä perustekniikoita, kuten reverse netcatin. Siinä kerrottiin toiseen tietokoneeseen yhdistämisen perusteet, kuten nmapin käyttäminen sen selvittämiseen, mitä käyttöjärjestelmää vastapuoli ajaa.

– Noin vuoden 2000 tienoilla aloin kiinnostua enemmän tietoturvasta. Tutkin asioita internetissä selaimella ihmetellen, miten eri asiat toimivat, ja aloin opetella HTML:ää ja TCP/IP:tä. Tein sitä ihan itsenäisesti.

Ensiaskleet

Ura hakkerina ei ollut vielä sinetöity. Vaikka Donnie ei sitä silloin tajunnutkaan, on mahdollista palata hetkeen, jossa kaikki alkoi valua siihen suuntaan.

Selaimella tehtyjen kokeilujen myötä hänen onnistui selvittää, miten erään aikuisviihdettä tarjoavan webcam-palvelun ohjelmisto toimii. Järjestelmästä löytyi vakavia puutteita, joiden avulla mallien IP-osoitteet sai kerättyä suoraan verkkosivulta.

– Pidin tiedot itselläni pitkään, mutta päätin sitten, että mallien yksityisyys on tärkeämpää. Windows-järjestelmässä oli siihen aikaan pahoja haavoittuvuuksia, joten IP-osoitteen tietämällä olisi ollut todella helppo murtautua heidän tietokoneisiinsa. Siihen ei vaa-dittu juuri mitään osaamista.

Donnie yritti ottaa yhtiöön suoraan yhteyttä ja tarjota konsultointia mak-sua vastaan. Hänen tarkoituksensa oli ennen pitkää saada tietoturva-aukko korjattua. Yhtiö ei kuitenkaan suostunut.

– Julkaisin sitten tiedot. Se pakotti heidät toimimaan, hän toteaa tyynesti.

Maine

Donniesta oli tullut hakkeri, ja pian

hän löysi muita kaltaisiaan.

– Tapasin Sub7-etähallintatroijalaisen takana olleen hakkeriryhmän ja päädyin aika syvälle heidän riveihinsä. Monet tuolloin mukana olleista ovat nykyisin hyviä ystäviäni. Jotkut ovat nykyään [tietoturva-]alan huippuja – ja heidän taustansa on se, että he hakkerivat muita, käyttivät takaportteja ja kaappasivat tietokoneita, hän kertoo huvittuneena.

Muiden hakkereiden tapaamisen

"...en ihan ymmärtänyt goto-silmukkaa."

myötä Donnie tutustui myös "full disclosure" -periaatteeseen ja Full

disclosure -postituslistaan, jonka tarkoituksena on pakottaa ohjelmistotoimittaja korjaamaan ongelmat ja tehdä ihmiset tietoisiksi niistä. Myös Donnie halusi tehdä jotain löytämilleen tietoturvaongelmille.

– Aiemmin vain julkaisin löydökseni ottamatta yhteyttä ohjelmiston kehittäjiin tai kehenkään.

– Kun löysin yhä enemmän ongelmia, päätin että haluan siitä tunnustusta. Tavallaan ajattelin, että rakennan itseleni siitä uraa, ja niin kävikin. Olin aika hyvä siinä, mitä tein.

Donnie niitti mainetta ja kerrytti CV:tään muun muassa löytämällä puutteita sellaisten suuryritysten kuin HP:n, Microsoftin ja Applen tuotteista. Hän myös auttoi korjaamaan löytämiään virheitä ja sai nimensä virallisiin päivityksiin. Maineen kasvamisen myötä hän saattoi ryhtyä työskentelemään itsenäisenä tietoturvakonsulttina lopulta ansaiten elantonsa erityis-taidoillaan.

Suomeen saapuminen

– Tein tietoturvakonsulttina keikkatöitä. Kahden viikon raadannasta sai kolmesta kuuteen tuhatta dollaria, mutta seuraavasta keikasta ei ollut koskaan tietoa. Jotain saattoi olla tulossa, mutta sattuikin jotain muuta ja pian huomasi, etten ollutkaan tehnyt mitään kolmeen kuukauteen, Donnie kertoo.

Kun laskut kasaantuivat ja seuraavasta työkeikasta oli jatkuva epävarmuus, halu luopua yrittäjän elämästä voimistui.

Muutama vuosi ennen Suomeen muuttamista Donnie oli opettanut Tallinnassa käytännön hakkerointia, pitänyt seminaareja ja luentosarjoja ja tutustunut sitä kautta alueeseen.



Saadessaan mahdollisuuden muuttaa Suomeen hän ei epäillyt kauaa. Töitä ei kuitenkaan ollut valmiiksi tiedossa.

– Hankin työpaikan vasta muuton jälkeen. Koska tulin maahanmuuttajana Suomeen, koin velvollisuudekseni hankkia töitä ja maksaa veroja.

– Avoimia työpaikkoja ei ollut, joten otin yrityksiin suoraan yhteyttä. Hain vain netistä sanoilla ”Helsinki computer security company”. Lähetin CV:n muutamaaan paikkaan, ja Nixu otti ensimmäisenä yhteyttä. Kävin siellä haastattelussa, mutta päädyin lopulta erääseen kansainväliseen telecom-yhtiöön.

Tällä hetkellä Donnie on toiminut nykyisessä työpaikassaan tietoturvakonsulttina noin kuusi vuotta. Tuoteosastolla ja tietoturvakäytössä työskentelemisen jälkeen hän siirtyi konsernin liiketoimintayksikköön, jossa hän tekee penetraatiotestejä, kesätyösuorvioita ja konsultaatiota.

Yritystason idiotismia

Vaikka Donnie työskentelee nykyisin suuryrityksen tietoturvakonsulttina, vapaa-aikanaan hän edelleen tutkii, opettelee uutta ja rikkoo asioita selvittääkseen, miten ne toimivat.

– Löysin hiljattain vian IoT-laitteesta. Käyttämäni tekniikka oli niin tyhmä, ettei sitä olisi edes pitänyt kokeilla. Löytämäni tietynlainen RFC:n vastainen URL-merkkijono mahdollistaa pääsyn kriittisiin järjestelmätiedostoihin.

Hän selittää, että koostamalla URLin tietyllä tavalla, RFC-sääntöjen

vastaisesti, tietystä laitteesta saa ilman kirjautumista ulos käyttäjätunnuksia, salasanoja ynnä muuta.

– Olin aiemmin nähnyt samanlaista tekniikkaa käytettävän vähän samantyyppisessä tuotteessa, joten kokeilin sen muunnelmaa, ja se toimi. Olen kokeillut sitä moneen muuhun laitteeseen, eikä se toimi niissä, joten todennäköisesti kyse on URL-käsittelyn toteutuksesta juuri tässä tuotteessa.

Donnien ei kerro suoraan, mikä laite on kyseessä, mutta kuvailee sitä pääosin kotiooloissa käytettäväksi, henkilökohtaiseksi intiimituotteeksi. Hän ei usko kenenkään muun vielä löytäneen samaa haavoittuvuutta. Laitteita on hänen mukaansa noin puoli miljoonaa, ja ne on kaikki yhdistetty internetiin. Yksi jälleenmyyjistä on yhdysvaltalainen.

– Yritin ottaa yhteyttä heihin ja selitin, mitä olin löytänyt. He kysivät, mikä tuote on kyseessä, koska niitä on muutamia erilaisia. Kuvailin sen heille, ja vastaus oli: ”Hei, tässä on linkki tuotteeseen. Katso lisätietoja tuotekuvasesta.” He siis hitto vie lähettivät minulle Amazon-linkin! hän tuhahtaa ja jatkaa ääneen sähköpostikirjeenvaihdon kertaamista tulostamaltaan paperilta. Jälleenmyyjän edustajien kanssa käyty keskustelu oli niin absurdi, että hän halusi jakaa sen työtovereilleen, ja tuloste sattuu olemaan hänellä yhä mukana.

– Vastasin heille:

”Ette tainneet nyt ymmärtää. En etsi jotain tiettyä tuotetta. Olen löytänyt useita tietoturvaongelmia, jotka koskevat kaikkia tuotteitanne. Niiden avulla tuote on mahdollista ottaa täysin hallintaan ja julkistaa asiakkaan tietoja.

Useimmat yhtiöt tarjoavat löydöksistä niin sanotun bugipalkkion, mikä on parempi vaihtoehto kuin se media-

huomio, joka seuraa tietojen julkistamisesta. Julkinen paljastus vaarantaa käyttäjien yksityisyyden ja voi johtaa siihen, että tuotetta ei enää käytetä.

Tuotteenne on tarkoitettu intiimisti ja yksityisesti käytettäväksi ja ihmiset toivovat voivansa luottaa siihen.”

– He puolestaan vastasivat: ”Kiitos kun kysyt tuotteestamme, tuote on helppo asentaa ja suojattu pääkäyttäjän salasanalla, joten tietoturvasta ei tarvitse huolehtia. Tuote on helposti varmistettu ja helppo asentaa.”

– Vastasin: ”Kiitos, mutta et nyt ymmärrä. Näytä tämä sähköpostikeskustelu tekniselle johtajalle.”

– Ja vastaus oli: ”Hei. Mitä tietoturvaan ja yksityisyyteen tulee, on todennäköisempää, että tuote hakkeroidaan, jos Wi-Fi on käytössä. Suosittelemme käyttämään salasanasuojattua Wi-Fi-yhteyttä. Alkuasetusten tekemisen jälkeen kannattaa mennä sovellukseen ja muuttaa käyttäjätunnus ja salasana. Alkuasetuksissa annetaan yleinen käyttäjätunnus ja salasana, jotka voi muuttaa myöhemmin. Muuttamalla ne voit

”Kiitos kun kysyt tuotteestamme, tuote on helppo asentaa ja suojattu pääkäyttäjän salasanalla.”



parantaa tietoturva. Voit myös muuttaa tuotteen aliaksen ja nimen, jolloin muut eivät tiedä, että kyseessä on tämä tuote.”

– Viimeisin vastaukseni oli: ”Valitettavasti et ymmärrä etkä välitä. Laitteessa on hyödynnettävissä oleva tietoturvaongelma. Se ei liity alkuasetuksiin, ja se voidaan ohittaa internetistä ilman mitään kirjautumista.”

Donnien äänestä kuuluu läpi epäusko ja turhautuminen, vaikka kirjeenvaihdosta on jo jonkin aikaa. Hän kertoo vielä saaneensa yhden viestin.

– Se oli: ”Pahoittelen, mutta emme oikein ymmärrä, mitä ajat takaa. Olemme selittäneet kaiken tietoturvajärjestelyihin liittyen. Lisäksi tuotettamme ei voi käyttää ilman laitetunnusta, joka on yksilöllinen.”

Donnien mukaan tunkeutumisen järjestelmä-tietoihin tapahtui jo pari kuukautta

aiemmin, vaikka hän pääsikin myös laitetunnuksiin käsiksi pari päivää sitten. Hän paheksuu kirjavasanaisesti tuotteen huonoa toteutusta. Tietojen julkaisu johtaisi kuitenkin siihen, että kymmenet tuhannet muut hyökkäisivät laitteen kimppuun. Donnie ei löydä syytä tehdä niin.

– Ainoa motivaatio voisi olla rahallinen korvaus. Haluaisin bugipalkkion. Myyjä ei selvästikään tunnusta, että heillä on tietoturvaongelma, eikä se todellakaan aio maksaa mitään bugipalkkiota.

Isossa-Britanniassa on tuotteen

toinen jälleenmyyjä, johon Donnie saattaa vielä yrittää ottaa yhteyttä. Todennäköisesti tapaus jää kuitenkin avoimeksi.

Valitettavasti tämä yhtiö ei ole ainoa yhtiö, jonka vastaus tietoturvaongelmaa koskevaan yhteydenottoon on ollut ala-arvoinen. Donnie kertoo saavansa samanlaisia vastauksia paljon, mutta toisaalta moni yritys myös vastaa asiallisesti bugilöydöksiin.

Lempisalaliitto

Kun Donnielta kysyy, onko hänellä jotain suosikkitarinaa tietoturvakatastrofiin liittyen, vastausta ei tarvitse odottaa: MS Blaster. Henkilökohtaisen kokemuksen jakaminen saa hänen silmänsä taas kiilumaan innostuksesta.

– Vuonna 2003 New Yorkissa ja koko itärannikolla oli iso sähkökatkos. Siitä syytettiin kaatuneita puita ja junan raiteita tai jotain sellaista. Kun kuulin siitä uutisista, ensimmäinen ajatukseni oli: hah, blastattu! Donnie nauraa.

– **Mikko Hyppönen** esintyy samassa dokumentissa kanssani ja sanoo suoraan, että uskoo niin juuri käyneen. Hakkeriyhteisön ulkopuolella sitä ei myönnetä, mutta me *tiedämme*, että MS Blaster katkaisi sähköt.

Hyökkäykseen tuskin oli mitään erityistä syytä, hän sanoo. Kehitysversion jälkeen Blasterista tehtiin mato, joka levisi IP-osoitealueiden perusteella. Saatuaan IP-osoitealueen yhdeltä koneelta se skannasi sen, laajensi

aluetta ja skannasi uudestaan. Mato ei valikoinut kohteitaan mitenkään.

Donnie kertoo joutuneensa madon kanssa tekemisiin jo varhain.

– Haavoittuvuuden löytäjät tai henkilöt, jotka saivat sen käsiinsä varhaisessa vaiheessa, käyttivät sitä ja tekivät sitten palvelunestohyökkäyksen minua kohtaan kaappaamalla koneilla. Sain sen selville vain, koska hakkeroinkin heidät, hän paljastaa hieman ylpeyttä äänessään.

Vasta pari viikkoa myöhemmin mato sai virallisen nimi ja sen olemassaolo tunnustettiin.

Tietoisuus

Sen sijaan, että tietoturvaongelmista syytettäisiin yksittäisiä käyttäjiä, Donnie haluaisi lähestyä tietoturva ohjelmistokehityksen puolelta.

– Tietoturva-asioissa koulutus on ensisijaisen tärkeää. Kehittäjien tietoisuutta tietoturvapuutteista pitäisi lisätä, mutta myös niiden ihmisten, jotka ovat vastuussa kehittäjien rekrytoinnista. Kehittäjien taitoja tietoturva-asioissa pitäisi testata nykyistä paremmin. Tietoturva on turha mieltä vasta jälkikäteen. Sen pitäisi kuulua kaikkeen tekemiseen, hän toteaa painokkaasti.

Vaikka Donnie on alansa huippu ja osaa kyllä rikkoa muiden koodin, hän ei ole itse koskaan tehnyt ohjelmistokehitystä. Se antaa hänelle omanlaista perspektiiviä tietoturva-asioiden nykitylaan.

– Minun oli vaikea päästä alalle, ja samalla näin kuinka ihmisille maksetaan hirveitä summia – ja sitten joku

minun kaltaiseni tulee ja pistää kaiken paskaksi. Niin ei saisi päästä käymään. Joku teki huonon päätöksen luottaessaan tähän tyyppiin. Johto on palkannut ihmisen, joka on antanut itsestään väärän kuvan. Ihmisen, joka tekee surkeaa koodia, hän paasaa ärsyyntyneenä.

Donnien mielestä kehittäjän on turhaa selittää puutteita sillä, että tietoturva-asioita ei koskaan opetettu koulussa, tai että tuotteesta ei ollut edes tarkoitus koodata turvallista. Hänen mukaansa tietoturva pitäisi oppia kielten tai kehitysympäristön kanssa.

– Jos valitsee tietyn ympäristön, siihen pitäisi perehtyä hyvin! Eikä vain siihen, vaan kaikkeen muuhunkin. Ei vain kehitykseen mutta myös siihen kehitykseen tai ympäristöön liittyviin tietoturvaongelmiin. Siitä sinulle maksetaan! Sitä varten olet opiskellut alaa!

Varmuus

Myös kehittäjien tekemä testaus on aihe, joka saa Donnien tunteikkaaksi.

– He testaavat tietoturvaongelmia, vanhentuneita päivityksiä ja väärää käyttöoikeuksia. Mutta samalla he eivät testaa esimerkiksi web-sovelluksen osia tai niiden syötteen käsittelyä. Pitäisi asennoitua niin, että ensin pitää tutkia vapaasti.

Donnien mukaan vapaa tutkiminen on hänelle luontaista, mutta on myös ihmisiä, joille se on vaikeaa. Hän kuvailee työskentelytapaansa hyppynä tuntemattomaan.

– Aika usein, kun teen jotain, mistä seuraa jotain odottamatonta, kehittäjä sanoo, että ei sitä ohjelmaa noin käytetä. Okei, selvä, mutta niin minä sitä käytin. Ja jos minä voin käyttää sitä niin, joku muukin voi. Joten ota se huomioon.

– Toinen juttu on, että kehittäjät tekevät tuotteen, kuten verkkopalvelun tai sovelluksen, ja sitä ajetaan tietyllä käyttöjärjestelmällä. Kun pyydät tietoturvatestien tuloksia, saat käytännössä tulokset käyttöjärjestelmän testauksesta. He eivät osaa räätälöidä testejä tai työkaluja omalle tuotteelleen. Jos käytetään perusasetuksia, useimmat työkalut testaavat käytännössä käyttöjärjestelmää.

Käyttöjärjestelmästä riippumatta hän käyttää mieluummin oikeita koneita virtuaalisten sijaan.

– Ostin tänään käytetyn Windows-

tabletin neljällä kymppillä. Ihan vain, koska siinä on Windows. Haluan käyttää muutamaa ohjelmaa, jotka toimivat vain Windowsissa. En pidä virtuaalikoneista. Pidän mieluummin kymmentä läppäriä kuin yhtä, jossa on yhdeksän virtuaalikonetta, Donnie hymähtää.

Hän selittää, että jos esimerkiksi testataan sitä, mitä pakeille tapahtuu verkossa, niin virtuaalikoneen käyttämiseen liittyy ongelmia. Kun paketit lähettää virtuaalikoneen sisältä, niiden pitää kulkea virtuaalisen ethernet-sovitimen läpi. Sama tapahtuu uudelleen, kun paketit palaavat.

– Jos teet matalan tason fuzz-testausta protokollalle, on vaikea tulkita oikein testien tuloksia, jos virtuaalinen pino ei ole täysin avoin ja standardien mukainen kuten oikea, fyysinen verkkokortti olisi. Joudut siis luottamaan täysin virtuaalisoinnin ajuritoteutukseen. Minulle se ei riitä.

Lannistuminen

Keskustelemme tietenkin myös Linuxista, johon monella hakkerilla on erityinen suhde. Donnie alkoi itse käyttää Linuxia jo vuonna 1993, ennen ensimmäistä virallista Red Hat -jakelua.

– Käytin Slackwarea. Näin tarjouskirjakaupassa Linux-kirjan, joka oli ainakin viisi senttiä paksu. Siinä oli komentohakemisto ja kaikkea sellaista ja mietin heti että se vaikutti siistiltä!

Sitten kaikkialla alkoi näkyä Linuxia ja internetiä. Donnie alkoi käydä Linux-käyttäjryhmien tapaamisissa, mutta kertoo, että niistä jäi paha maku suuhun.

– Kaikki keskittyivät vain siihen, mis-



tä olivat superkiinnostuneita ja minkä osasivat erittäin hyvin, ja kun huomaisivat että toinen ei osaa eikä tiedä siitä mitään, he eivät muusta puhuneetkaan. Ilmapiiiristä tuli tosi huono, etenkin siellä, missä itse vaikutin.

Hän sanoo lopettaneensa Linuxin ja internetin käytön kokonaan parin-kolmen vuoden ajaksi – juuri kun muut alkoivat intoilla siitä.

– En ole sellainen tyyppi, joka lähtee mukaan villityksiin, hän toteaa käyttäen englantinkielistä *fad*-sanaa.

Hän kertoo käyttävänsä nykyisin sekä Linuxia että Windowsia mutta pääkäyttöjärjestelmänään Windowsia. – Kun pitää saada asioita tehtyä, käytän Windowsia. Säättämiseen käytän Linuxia. Kun testaan jotain aktiivisesti, teen sen yleensä Linux-ympäristössä. Nykyinen työkoneni ei tue kaikkia käyttämiäni työkaluja, mutta useimmat työkalut löytyvät molemmille alustoille.

Uteliaisuus

Hairahdumme sivupoluille ja palaamme Donnieen ja hänen hakkerikaverihinsa. Jotkin asiat yhdistävät heitä kaikkia. Donnie kuvaa sitä leikkimiseksi.

– Se pitää meidät virkeinä, ja [teemme sitä], koska olemme uteliaita. Meistä on kiva tietää, että voimme rikkoa jotain, minkä tekemisestä sait paljon rahaa.

Uteliaisuus, leikkiminen ja innostus tietotekniikkaan tuntuvat olevan kantavia teemoja hänen tarinoissaan. Kaikesta, mitä Donnie kertoo, huokuu myös omanlainen katsontakanta maailmaan ja tapa lähestyä asioita. Kun kysymme, onko hänen ajattelutapansa taustalla tietynlainen hakkerimentaliteetti, hän vastaa epäsuorasti:

– Vain toinen käsistäni toimii, joten joudun koko ajan miettimään, miten teen jotakin. Miten kävelen ovelle kahden ruokakassin ja avaimien kanssa samalla, kun pidän kannettavaa kaimalossa. Joudun jatkuvasti ajattelemaan. Muiden ei tarvitse miettiä samalla tavalla. He vain käyttävät käsiään. Joudun itse aina miettimään, miten kättäni.

Hakkerointia ei motivoi Donnien

kohdalla jokin aate, vaikka monelle hänen kaltaiselleen harmaahatulle, laitto- man ja laillisen rajamailla liikkuvalle hakkerille, hakkerointi on aktivismia.

– Menin erääseen hakkerikonferenssiin Yhdysvalloissa. Se oli hyvä konferenssi, mutta käytännössä pelkkää hacktivismia, eli paljon pyrkyreitä, jotka ihailevat muutamaa alalle päässyttä.

Hän ei ole itse kiinnostunut ”taistelemaan systeemiä vastaan”, vaan hänelle

hakkeroinnissa on kyse jostain aivan muusta.

– Olen enemmän tekninen ihminen, samoin kuin monet tuntemani hakke-

rit.

Hän myös kokee, että esimerkiksi kyberrikollisten harjoittama toiminta on enemmän työtä kuin se, mitä hän tekee. Hän ei usko rikollisten hallitsevan kovin monimutkaisia tekniikoita, vaan toimivan lähinnä heille opettujen keinojen puitteissa. Heitä ohjaa raha, ei uteliaisuus.

Donnie siirtyy puolustuskannalle, kun kysymme hänen omasta aiemmas- ta mustahattuilustaan.

– Olen mielestäni enemmän harmaakuin mustahattu. Voin sanoa, että olin aiemmin mustahattu, koska tein mitä halusin. Mutta minullakin oli rajani. En hyökännyt NATO-maihin enkä korkeakouluihin. En ollut pahantah- toinen, ja vaikka dumppasin tietokan- toja, en tehnyt tiedoilla mitään.

– En koskaan ajatellut olevani mustahattu, koska en tehnyt mitään rahalli- sen hyödyn vuoksi.

Vastuu

Olemme tyytyväisiä äänittämiimme tarinoihin, mutta haluamme vielä tietää, olisiko Donniella jotain, mitä hän haluaisi jakaa lukijoille. Hän miettii hetken, ja pidämme pienen tauon, jonka jälkeen painamme taas punais- ta nappia matkapuhelimen näytöltä ja jatkamme kuuntelemista.

– Tietoturvassa on kyse perspektiivis- tä, näkökannasta asioihin. Mieti, antaisitko äitisi ja ihmisten, joista välität, käyttää tekemäsi tuotetta. Jos antaisit, asiat ovat todennäköisesti hyvin. Jos et ole varma, tee sialle jotain. Pyri oppi- maan lisää.

Donnie hautautuu taas hetkeksi aja- tuksiinsa.

Hän kehottaa kysymään työnanta- jalta, onko mahdollista saada lisäkou- lutusta taitojensa kehittämiseen. Omat heikkoutensa täytyy hänestä voida tunnustaa, jotta taitojaan voi kehittää.

Hän penää kehittäjien vastuuta omasta työsuorituksestaan.

– Saattaa olla, että pärjät, koska ke- hittäjiä tarvitaan ja osaat monia asioi- ta, mutta työnantajasi on jo käyttänyt paljon rahaa integroidessaan sinut yri- tykseensä. Se on iso investointi, joten aivan varmasti työnantaja myös mak- saa SANS-kurssin tai muun vastaavan laajemman kurssin, jos siitä on työn kannalta etua. Voit myös hyödyttää koko tiimiäsi sen kautta.

– Kerro siis avoimesti, että välität sii- tä mitä teet ja pidät työstäsi, mutta jotkin asiat huolettavat. Työnantajasi arvostaa sitä ja maksaa sinulle kurssin käymisestä. Kanna vastuusi ja paljasta omat heikkoutesi.

Hänen mukaansa epäonnistumista ei tule pelätä, ja hän korostaa, että 99 prosenttia ajasta hän itekin epäonnis- tuu. Myös tilanteissa, joissa muut on- nistuvat.

– En olisi koskaan löytänyt aiemmin mainitsemaani haavoittuvuutta, jos en olisi vain päättänyt kokeilla jotain. Sitä ei olisi edes pitänyt kokeilla, ja yllättä- en se toimikin.

Ylpeys

Aurinko alkaa laskea ja Ruoholahden katuvalot särivät päälle yksi toisensa jälkeen. Donnie laskee tyhjän olutlasin pöydälle, ja valmistaudumme lähte- mään.

Haastattelustamme on jäänyt pyöri- mään mieleen paljon erilaisia ajatuk- sia, ja digitaalinen tallenne keskuste- luistamme sujahtaa kännykän mukana taskuun odottamaan ruotimista. Pääl- limmäiseksi tunteeksi jää ehkä se, että Donnie on todella ylpeä tekemästään työstä ja suhtautuu siihen intohimoi- sesti.

– Työnantajani on tällä hetkellä maa- ilman suurin kriittisen viestintäinfra- struktuurin toimittaja. Maailman ja valtioiden turvallisuus riippuu meidän laitteistamme. Teen kaiken voitavani varmistaakseni niiden tietoturvan, ja sanon sen erittäin vakavissani. Kyse ei ole niinkään uskosta yhtiöön vaan toimialaan, joka tuottaa kriittistä inf- rastruktuuria, hän toteaa painokkaasti.

– Tätä minä teen. 🐙



Viikko tietoturvasekatyömiehen elämästä

Äitini opetti minulle aikanaan, että leipä on pieninä palasina maailmalla.

Mikko Hyppönen,
tutkimusjohtaja, F-Secure Oyj

TIISTAI 9.5.2017

Ajelen aamulla F-Securen pääkonttorille Helsingin Ruoholahteen. Ohjelmassa on muutama tapaaminen ja sähköpostin purkua. Päivällä käyn pitämässä keynote-puheen pilvipalveluihin keskittyvässä seminaarissa hotellissa Helsingin keskustassa. Kun pilviveikot on peloteltu, palaan konttorille isännöimään asiakasryhmän vierailua laboratoriossa. Siellä olevaa Faradayn häkkiä on aina hauska esitellä.

KESKIVIIKKO 10.5.2017

Lähden aamulla Messukeskukseen pitämään Prosessipäivien aloituspuheenvuoron. Puhe loppuu klo 10.00, ja viidessä minuutissa olen taksissa matkalla kohti 11:05 lähtevää lentoa Osloon. Siellä on alkamassa yksi Pohjoismaiden kovatasoisimmista hakkerikonferensseista, Paranoia.

Keskiviikon keynote vetää Charlie Miller, maailman tunnetuin autohakkeri. Tunnen Charlien jo vuosien takaa, ja rupattelemmme iltapäivällä pitkään. Charlie mainitsee, että hänen ja Chris Valasekin löytämät aukot aiheuttivat Chryslerille 14 miljardin dollarin kustannukset.

”Jos Chrysler olisi maksanut meille 10 miljardia, emme olisi kertoneet kenellekään löydöstä, ja Chrysler olisi säästänyt 4 miljardia!”

Konferenssin jälkeen puhujat koontuvat päivälliselle. Pöytäkunnassa istuu ainakin kaksi entistä tiedustelupalvelu NSA:n työntekijää.

TORSTAI 11.5.2017

Pidän aamun ensimmäisen keynoten Paranoiassa. Kyseessä on kymmenes vuosittainen Paranoia. Puhun siitä, miten paljon lyhyessä ajassa on muutunut. Totean, että 10 vuotta sitten meillä kaikilla oli suomalainen puhelin taskussa ja nyt kellään meistä ei ole. Aika lentää, varsinkin netissä.

Session jälkeen siirryn heti Twitterin ääreen, koska tiedän, että F-Secure on aamulla julkaissut pörsstitiedotteen: osumme englantilaisen tietoturvayhtiön. Levitän uutista Twitterissä ja toivotan uudet työntekijät tervetulleeksi taloon.

Suuri seuraajamäärä on opettanut varovaiseksi. Aina joskus käy niin, että linkkaan Twitterissä sivustoon, ja se menee heti nurin, koska niin moni vieraillee siellä yhtä aikaa.

Hoitelen Twitteriä taksista, koska puolilta päivin lennän Tukholmaan IDG:n vuotuisen IoT-tapahtumaan. Minua ennen puhuu Volvon nettiyhteydet suunnitellut nainen ja porojo internetiin kytkevä mies. Puhun IoT:n tulevaisuuden hyvistä ja huonoista puolista. Vierailu Tukholmassa jää lyhyeksi. Singahdan heti puheen jälkeen

Arlanda Expressillä kentälle ja lähdän iltakoneella Madridiin.

PERJANTAI 12.5.2017

Madridissa puhun aamulla noin 150 asiakkaalle suuryritysten tietoturvaratkaisuista. Puheen ja paluulennon välillä on muutama tunti aikaa. Palaan kentälle tekemään töitä lentokenttä-loungessa.

Tuskin ehdin istahtaa, kun puhelin soi. Konttorilta Suomesta kehoitetaan ottamaan yhteys yhteen suurimmista asiakkaistamme Espanjassa. Soitan johtoryhmän jäsenelle, ja selviää, että heidän verkossaan on tuhansia tartuntoja uudesta haittaohjelmasta. Mikä pahempaa: tartunta leviää yhä.

Kyseessä on uusi tiedostoja salaava lunnastroijalainen – WannaCry. WannaCry leviää kuin kulovalkea ja muuttuu päivässä historian suurimmaksi lunnastroijalaisepidemiaksi. Uhreina on lähes pelkästään suuryrityksiä, ja saastuneita koneita on illalla jo melkein 200 000.

Puhelimeni soi tauotta, ja ehdin vielä koneessa puhua monta puhelua, koska lähtö kestää. Välipysähdyskentällä Frankfurtissa on ukkosmyrsky. Ehdin kuitenkin Helsingin lennolle, koska sekin on myöhässä. Laskeudun kotikentälle vähän ennen kolmea lauantaiamuna.

LAUANTAI 13.5.2017

Ehdin nukkua vajaat kuusi tuntia ennen kuin WannaCry-ruljanssi jatkuu. Aamuun mennessä tiimimme on purkanut WannaCryn koodin kokonaan. Selviää, että WannaCryn leviäminen on pysähtynyt, koska tuttu brittitutkija on löytänyt koodista toiminnon, jonka avulla hän sai epidemian globaalisti pysäytettyä. Mitalin arvoinen tempku.

Teen Skypea kautta kotoani kaksi live-haastattelua BBC:lle. Toinen niistä menee BBC Worldille, jolla on kymmeniä miljoonia katsojia. Haastattelu menee hyvin. Tartunnan saaneita yrityksiä on kaikkialla maailmassa: sairaaloita, autotehtaita, voimalaitoksia, junayhtiöitä...

SUNNUNTAI 14.5.2017

Äitienpäivä menee WannaCryn kanssa painiessa. Anteeksi Äiti.

MAANANTAI 15.5.2017

Maanantaina selviää, että Wannacryn tuhot Aasiassa ovat paljon luultua suuremmat. Haittaohjelman tartuntoja yritysverkoissa ei ehditty huomata ennen viikonloppua. Puhelin soi ja soi. Asiakkaat ja media soittivat.

Nauhoitan parin minuutin yhteenvedon tilanteesta ja vastaan radioasemien haastattelupyynnöihin maailmalle heille OGG-tiedoston. Ainakin eteläafrikkalainen ja itävaltalainen radioasema toistavat tiedoston kuulijoilleen sellaisenaan.

CNN soittaa ja haluaa tehdä haastattelun Skypea kautta liveinä maailmanlaajuiseen jakeluun. Haastattelijana on CNN:n naistoimittaja Hong Kongista. Olen tavannut hänet vuosien varrella monta kertaa. Ennen suoran lähetyksen alkua muistelempa haastatteluja muun muassa Sony Rootkitista vuonna 2005 ja Confickerista vuonna 2008. CNN:n jälkeen hyppään taksiin Helsinki-Vantaalle.

Nyt istun Finnairin koneessa matkalla Barcelonaan. Siellä alkaa huomenna F-Securen tärkein vuosittainen asiakastilaisuus, jonne keräämpa operaattori-asiakkaitamme kaikkialta maailmasta. Ei ole vaikeaa arvata, että puheenaihe numero yksi tulee olemaan WannaCry.

Barcelonasta matkani tulee jatkuamaan Haagiin, puimaan WannaCrya Europolin tutkijoiden kanssa.

Leipä todella on pieninä palasina maailmalla. Myös tietoturva-alalla. 🍞



RuuviTag – Suomalainen majakka esineverkoille

Esineverkolla (*Internet of Things*, IoT) viitataan perinteisistä tietokoneista poikkeavien esineiden kytkeytymiseen internetiin. Alkujaan siinä oli kyse tavaroiden tunnistamisesta tageilla, kuten RFID:llä. Esineverkot palasivat juurilleen, kun Nokian kehittämä Wibree integroitiin Bluetooth 4-standardiin Bluetooth Smartin tai Low Energy (BLE) nimellä. Yksi BLE:n ominaisuus on majakka.

BLE-majakka kuuluttaa yksilöllistä tunnistettaan läheisille BLE-laitteille. Applen iBeaconin kaltaista protokollaa käytettäessä älylaite reagoi majakkaan, jos siihen on asennettu yhteensopiva sovellus. Sovellus saa majakalta tunnisteen ja mahdollisesti paikallista anturitietoa. Googlen julkaisema Eddystone-protokolla tukee myös fyysistä webiä (*Physical Web*), jossa majakka kuuluttaa standardia web-osoitetta. Majakka ei kuitenkaan ole palvelin, joten lisätoiminnot vaativat aina internetiä.

Tähän markkinarakoon iskee suomalaisen Ruuvi Innovations Oy:n RuuviTag. Se joukkorahoitettiin viime vuonna, ja ensipainos toimitettiin keuhkatalvella tukijoille – myös Skrollille. RuuviTag toimii Eddystone/iBeacon-majakkana (sis. Bluetooth 5 -valmius) ja versiosta riippuen tagissa on muun muassa lämpötila-, ilmankosteus-, ilmanpaine- ja kiihtyvyyssanturit. Paristo kestänee vuosikymmenen ja radiokantama pellolla on jopa kilometrin.

Skrollin testissä kaupunkioiloissa Android-luuri kuuli majakan automaattisesti muutaman kymmenen metrin etäisyydeltä ilman mitään asennuksia, ja sääasemakin aukesi klikkauksella selaimeen. Toimii. Varsinainen juju on kuitenkin tuunattavuus – RuuviTag taipuu uusiin sovellutuksiin avoimella kehitysympäristöllä. Lisätietoja: ruuvi.fi 🍞

Janne Sirén



NOKIA 3310

Tasavallan matkapuhelin

Nokia 3310 kuuluu saunan, sisun ja rallin ohella asioihin, joista Suomi tunnetaan. Skrolli tutustui kansankännykkämme matkaan vuosituhannen vaihteen tusinaluurista 2010-luvun ilmiöksi.

Teksti ja kuvat: Janne Sirén

Kun ulkoministeriö julkaisi Suomelle omat emoji-t vuonna 2015, ensimmäiset kolme olivat sauna, hevuri ja Nokia 3310 -kännykkä. Viimeinen oli kuitenkin alkuaan varsin epätodennäköinen sankari. Julkaisuvuonna 2000 hohdokkaat simpukkapuhelimet kilpailivat mielikuvista ja älypuhelimet olivat ovelia. 3310 oli juro, juhannustaivaan värinen patukka, josta oli design kaukana.

Siinä ei ollut mitään erityistä. 3310 ei ollut ensimmäinen nokialainen helpokäyttöisellä Navi-näppäimellä (se oli 3110), ei ensimmäinen matopelillä (6110) tai helposti vaihdettavilla kuorilla (5110). Siinä ei ollut wappia (7110) tai värinäyttöä (3510i). Se ei myynyt maailman eniten (1110) eikä aikalaisistaan eniten (3210). Se ei edes synnyttänyt Nokia-puhelinten kestävyysmeemiä (1100). Sitä paitsi se taidettiin suunnitella Tanskassa, vaikkakin valmistettiin Suomessa.

Amerikkalainen tietokonevalmistaja Commodore onnistui aikanaan luomaan samankaltaisen ilmiön. Kaikki Euroopassa muistavat Commodore 64:n, mutta harva muistelee numeroita 4, 16, 20, 128 tai edes numeroa 500 – saati useimpia kilpailijoita. Vaikka myyntiluvuilla on osansa, oman aikansa symboliksi nouseminen vaatii ennen kaikkea oikeaa ajoitusta. Miksi juuri Nokia 3310 on noussut kännykkäajan ikoniksi?

Peruskännykän joutsenlaulu

Nokia 3310 syntyi ajankohtaan, jossa kiihkeästi kasvanut matkapuhelinmarkkina oli värinäyttöjen ja data-liikenteen murroksessa. Internetiä luuriin vasta kosiskeltiin, ja kamerapuhelin oli vielä parin vuoden päässä. Tätä edeltäneet, vuosittain vaihtuneet massasuosikkimallit pirstaloituivat pienemmiksi puroiksi eri suuntiin. 3310 jäi kantamaan peruspuhelimen lippua jopa viideksi vuodeksi.

Vanha Nokia 3310 oli 1990-luvun matkapuhelinevoluution puristama timantti. Puhelinmuistio, kourallinen soittoaaniä värinäitoiminnolla ja useampi päivä akkuvirtaa soittojen odotteluun. Pieni mustavalkonäyttö, johon mahtuivat ennakoivalla T9-näppäimistöllä kirjoitettavat ylipitkät tekstiviestit. Käyttäjän huomiosta kilpaili vain neljä sisäänrakennettua peliä.

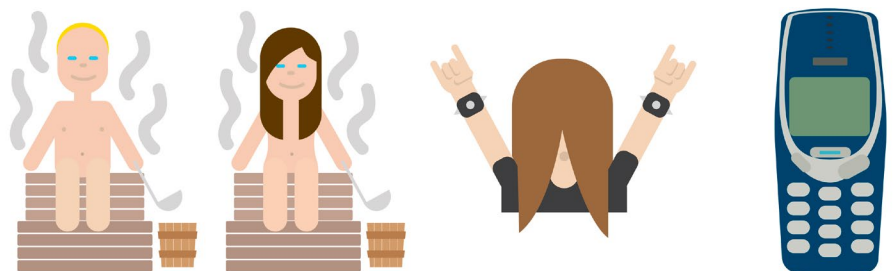
Kuten Commodore 64:stä aikanaan, 3310:sta tuli klassikko osittain siksi,

että se oli tarpeeksi halpa, tarpeeksi hyvä ja lajinsa viimeinen volyymituote. Se oli läntisen maailman kuvakulmasta viimeinen kerta pitkään aikaan, kun massat ostivat yhtä puhelinta yli sata miljoonaa kappaletta – seuraava kerta tulisi vasta vuosia myöhemmin Applen kuorissa. 3310 oli monelle se viimeinen peruspuhelin.

Nokia 3310 edusti myös katoavaa valmistustapaa, jossa kutistunut elektroniikka – etenkin herkätkä akku ja näyttö – ympäröitiin väljällä muovikuorella. Puhelimen pudottaminen oli tuohon aikaan yleisesti lähinnä kosmeettinen haitta, mutta suurenevat näytöt, tiivistetyt kotelot ja kasaan liimatut arvomateriaalit tulisivat pian muuttamaan tämän. 3310:lla oli tuuria, sillä tämä murros liitettäisiin juuri siihen.

Tuhoutumaton Nokia

Oikeastaan koko Nokia 3310 -ilmiö on huijausta. Jos menneestä kännykkäpe-



Ensimmäiset Suomi-emoji-t: Sauna, Headbanger ja Unbreakable. (Ulkoministeriö, 2015.)

rinteestä pitäisi poimia kovin sankari, se olisi Nokia 1100. Kolme vuotta 3310:n jälkeen kehittyville markkinoille julkaistu malli myi enemmän (sisämallinsa 1110:n kanssa tuplasti enemmän) kuin 3310. 1100:n pölynkestävä näppäimistö jaksoi myös jatkaa vielä, kun 3310 oli jo tukossa.

1100 oli jopa lähtölaukaus 3310:n toiselle tulemiselle, internet-ilmilölle tuhoutumattomasta Nokiasta. Vuonna 2011 Reddit-sivustolle lähetettiin kuva Tuomiovuoresta ainoana paikkana, jossa Nokia 1100 voidaan tuhota – rinnastaen se *Taru sormusten herrasta* -kirjasarjan lähes tuhoutumattomaan mahtisormukseen. Eräässä kuvan saamassa kommentissa 3310 nimettiin saman kirjan Noitakuningaskaaksi. Noitakuningaskin oli vaikea tappaa, mutta tuhoutui silti kirjassa ennen mahtisormusta.

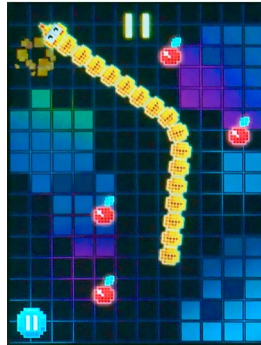
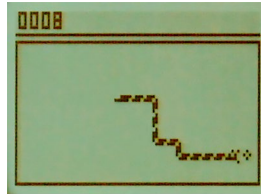
3310-meemipallo oli kuitenkin päästetty irti, ja Tuomiovuorta seurasi toinen toistaan kekseliäämpiä kierto-kuvia nimenomaan 3310:stä milloin luodinkestävänä liivinä, milloin lattian tuhoajana. Tästä ulkoministeriömmekin innostui. Lopulta Nokia 1100:lla ei ollut mahdollisuuttakaan. Se oli lähes tuntematon länsimaissa, kun taas 3310:n muistivat kaikki. Legenda oli syntynyt.

Viraali-ilmioon tarttui myös Nokia yhdessä kumppaninsa HMD Globalin kanssa, joka lisensoi Nokian matkapuhelinbrändin vuonna 2016. Kun ”kaikki oli ennen paremmin” kiteytyy ilmiön myötä Nokia 3310:aan, nostalginen markkinarako on selvä. Ripotellaan päälle vielä ripaus trendikästä *digital detoxia*, niin ideaa voi yrittää myydä millenniaaleillekin.

Uusi Nokia 3310

”Nokia 3310” palasi touko-kesäkuun vaihteessa kauppoihin 59 euron hintaan. Kyseessä on kevyesti viritetty Series 30+ -sarjan kiinalainen Mediatek-peruspuhelin, jonka ulkonäkö on tyyliä lainattu 3310:lta. Ennakoimme tarinalla noloa loppua, mutta uskaltauduimme kokeilemaan.

Ensimmäkin uusi 3310 on vain kolhampi design-versio puolet halvemmasta Nokia 150:sta. Lisähinnalla saa väritettyä retromuovia ja huonon kamerasurkean sijaan. Nokia 150 olisi näistä se järkevempi ostos. Toisaalta vaikka somet ja netit laittaisi katkolle,



Vanha Snake II ja uusi Snake (alla).

puhelimien käyttö on nykyisin paljon muutakin. 3310:lla ei makseta pysäköintiä, navigoida, mitata urheilusuoritusta, ohjata kodinkoneita tai tilata maailmalla Uberia. Verkkotaajuudet eivät edes toimi kaikilla kaukomailla.

Vanha Nokia 3310 oli myös harkittu kokonaisuus, jossa oli vain perusasiat. Tässä Nokia/HMD:n rohkeus ja väivannäkö on loppunut kesken. Kenties kehittyvien markkinoiden takia ei ole uskallettu karsia uudesta 3310:sta täysin puhdasta peruspuhelinta, vaan huumorikärkiä on jätetty: Opera Mini -selain ei kuulu 2010-luvulle, eikä rautaa vain riitä (video)kuvaamiseen. Etenkin vanhan 3310:n rinnalla huomaa, kuinka uuteen on kertynyt kaikenlaista vähemmän loogista krääsää.

Mutta unohdetaan epäolennaiset kymppien hintaerot, mobiiliselain ja kamera. Ei tänne niiden takia tultu. Kuvitellaan, että 3310 on vain peruspuhelin, jolla voi lisäksi kuunnella radiosta ja muistikortilta musiikkia sekä käyttää samaan aikaan kahta eri micro-SIMiä. Olennaisempaa tosin on hankkia kaksi rinnakkaista SIMiä: yksi älypuhelimien ja yksi kakkosluuri-3310:aan.

Nyt alkaa jo hymyilyttää. Uusi 3310 näyttää nätiltä, paremmalta kuin edeltäjänsä. Siirtymät perusasioissa ovat salamannopeita, ja näyttökin on yllättävän hyvä. Puhelut ja tekstarit toimivat. Nokia-soittoäänensä on tunnelmaa. Musiikki soi kuulokkeilla passelisti. 3310 on myös mainio elektronikkapeli – matopeli viihdyttää, ja



Alkuperäinen Nokia 3310 (vas.) ja uusi. Mattasinisen (oik.) ja harmaan lisäksi uuden saa kiiltävän keltaisena ja punaisena. Kiitos Viestintätoimisto Paja ja HMD keltaisesta lainapuhelimesta.

lisää pikkupelejä saa Opera-kaupasta. Vaihdeettavan akun valmiusaika on kuukauden luokkaa. Rakennekin tuntuisi sietävän osuma.

Yllättäen uutta 3310:aa voi jopa suositella, jos oikeasti tarvitsee johonkin pelkkää puhelinta ja musiikkisoitinta. 3310 voisi olla tukeva-akkuinen kaveri kesämökille tai vaellukselle, ehkäpä myös radio juoksulenkille. Kännykkä sinne missä muuten jättäisit puhelimen kotiin. Työpaikan päivystysluurina se toimii myös erinomaisesti. Anoppi kuitenkin pysyköön Dorossaan, hänelle 3310:ssa on valikkoja puolet liikaa. Ja unohda se digivieroitus – retroilijalle tämä vitsi käy pian vanhaksi.

Nokia 3310:n tarina muistuttaa sanonnasta: älä tapaa sankareitasi. 3310:n sankaritaru on myytti, joka elää parhaiten muistoissa. Harvinaiseen tositärpeeseen olisi kuitenkin taas tarjolla kiva peruspuhelin. 📞



”Indestructible Nokia 3310” – yksi monista internetiä kiertävistä kunnianosoituksista.



Ystävämme Petri

Kun normaali ohjelmistotestaus ei riitä, on aika ottaa esiin täsmälliset matemaattiset mallit. Yksi näistä on Petri-verkot, joilla on mahdollista todistaa, että järjestelmä toimii oikein.

Teksti: Antti Ylikoski Kuvat: Toni Kortelahti, Nasu Viljanmaa

Ohjelmointi on vaikeaa, ja bugit ovat väistämättömiä, koska ihminen on erehtyväinen. Järjestelmän monimutkaisuus voi kasvaa päätä huimavaksi, kun ohjelmoidaan rinnakkaisia ja hajautettuja järjestelmiä, joissa useat säikeet kilpailevat yhtä aikaa samoista resursseista.

Monet turvallisuuskriittiset sovellukset, kuten lentokoneet ja ydinvoimalat koostuvat useista reaaliaikaisista systeemeistä, joiden on kommunikotava keskenään. Yksikin bugi saattaa tappaa satoja ihmisiä tai aiheuttaa ympäristökatastrofin. On myös kriittisiä tietoliikenneprotokollia, joita käyttävät miljoonat ja taas miljoonat laitteet päivittäin. Näiden suunnittelussa yksikin virhe tai haavoittuvuus voi tulla hyvin kalliiksi.

Näiden kriittisten järjestelmien suunnittelussa avuksi voi ottaa järeät työkalut – matemaattiset menetelmät, joilla voidaan etsiä järjestelmästä bugit tai jopa todistaa järjestelmä bugittomaksi matemaattisesti. Järjestelmä mallinnetaan matemaattisesti, ja sen käyttäytymistä voidaan

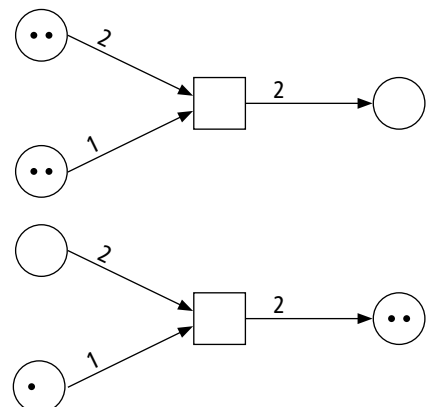
tutkia olemassa olevilla työkaluilla. On kehitetty useita matemaattisia formalismeja, jotka ovat tyypillisesti jonkinlaisia tilakoneita. Yksi näistä formalismeista on nimeltään Petri-verkko.

Matemaattinen malli

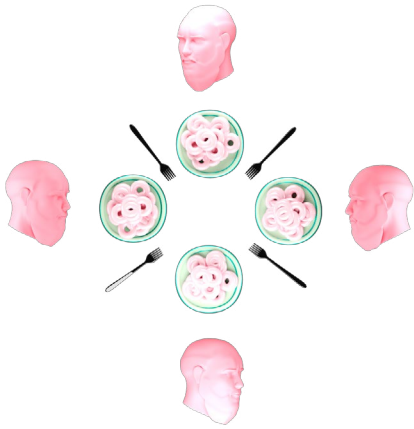
Carl Adam Petrin väitöskirja *Kommunikation mit Automaten* loi uuden tieteenalan – akateemisen Petri-verkkojen tieteen. Petri-verkkoja voidaan käyttää erilaisten rinnakkaisten ja hajautettujen järjestelmien ja kommunikoinnin mallintamiseen ja analysointiin. Niiden laskentavoima on pienempi kuin Turingin koneiden, mikä mahdollistaa niiden analysoinnin tietokoneella.

Matemaattisesti Petri-verkko on suunnattu kaksijakoinen verkko. Siinä on olioita ja suunnattuja kaaria, jotka johtavat oliosta toiseen olioon. Kaksijakaisuus tarkoittaa sitä, että oliot jakautuvat kahteen ryhmään ja kaikki kaaret kulkevat ainoastaan yhdestä ryhmästä toiseen ryhmään. Ensimmäisen ryhmän olioita kutsutaan paikoiksi ja toisen ryhmän olioita siirtymiksi. Paikoissa voi olla merkkejä (tokens), jotka voivat siirtyä paikasta toiseen siirtymien kautta.

Petri-verkko toimii seuraavasti: Siirtymillä on joukko tulopaikkoja (input places) ja joukko lähtöpaikkoja (output places). Kun siirtymän tulopaikoissa on riittävän monta merkkiä, niin sanotaan, että siirtymä on virittynyt. Tällöin siirtymä voi laukea. Kun siirtymä laukee, lähtöpaikoista otetaan pois ja tulopaikkoihin tuodaan tietty lukumäärä merkkejä.



Kaavio 1. Kuvissa paikat ovat ympyröitä, siirtymät neliöitä ja merkit mustia palloja. Kaaren kohdalla oleva numero kertoo, kuinka monta merkkiä kaaren läpi kulkee kerrallaan. Ylemmässä kuvassa on tilanne ennen siirtymän laukeamista, ja alemmassa kuvassa tilanne siirtymän laukeamisen jälkeen.



Kuva 2. Ruokailevat filosofit tarvitsevat kaksi haarukkaa syödäkseen.

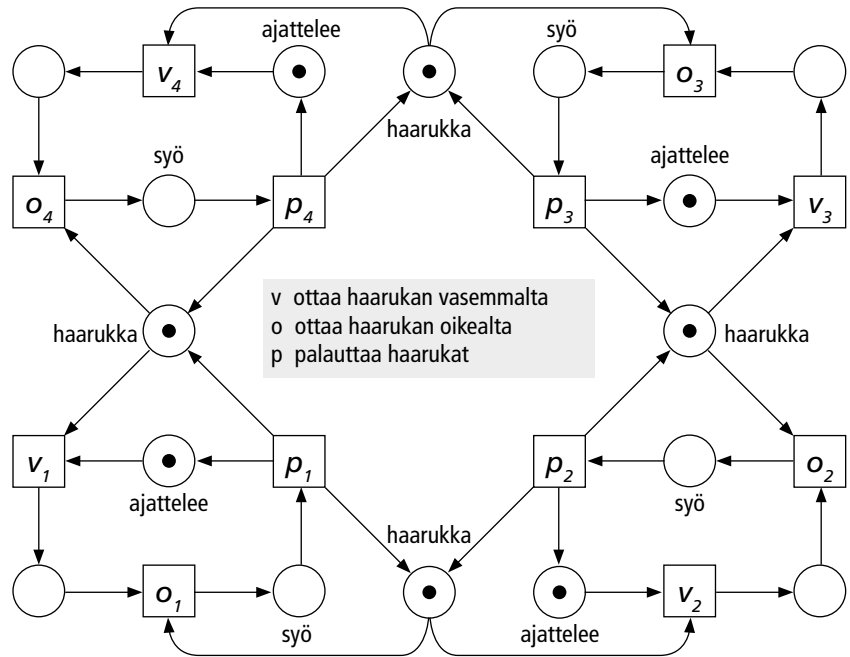
Tämä toiminta vastaa asiaa, joka tapahtuu usein todellisuudessa: kun määrätty ehdot ovat täyttyneet, niin jokin tietty tapahtuma voi tapahtua. Esimerkiksi matkakortinlukija voi veloittaa kortilta hinnan vasta, kun kortti on lukijan edessä, haluttu matka on valittu ja kortilla on tarpeeksi rahaa. Siirtymä poistaa sitten kortilta matkan hintaa vastaavan määrän merkkejä ja tuottaa merkin paikkaan, joka kuvastaa sitä, että kortilla on voimassa oleva lippu.

Petri-verkoilla voi mallintaa ja analysoida suhteellisen yksinkertaisia hajautettuja ja rinnakkaisia järjestelmiä ja fysikaalisia olioita, kuten esimerkiksi kerrostaloa hisseineen ja asukkaineen tai mekaanista kelloa.

Filosofit ruokapöydässä

Eräs kuuluisa tietojenkäsittelytieteen esimerkkiongelmia, jota voi kätevästi mallintaa Petri-verkoilla, on niin sanottu lounastavien filosofien ongelma. Tätä esimerkkinä käytettyä ongelmaa voi kuvata seuraavasti.

Joukko filosofeja elää elämää, jossa itse kukin filosofi vuorotellen ajattelee ja ruokailee. Filosofit ruokailevat pyöreän pöydän ääressä ja syövät aina spagettia. Jokaisen vierekkäisen filosofin välissä on yksi haarukka (ks. Kuva 2), ja ruokaillakseen kukin filosofi tarvitsee kahta haarukkaa. Tehtävä on synkronoida filosofien ruokailu ilman, että kukaan näantyy nälkään, koska viereissä istuvat filosofit omivat haarukat



Kaavio 3. Neljää ruokailevaa filosofia mallintava Petri-verkko. Paikat ovat ympyröitä, neliöt siirtymiä ja mustat pallot merkkejä. Kaarten painoja ei ole merkitty kuvaan, koska jokaisen kaaren paino on yksi.

itselleen.

Tämä esimerkkiongelmia havainnollistaa niin sanottua käyttöjärjestelmän deadlock-tilannetta. "Deadlock" tarkoittaa sitä, että useampi prosessi haluaa käyttöönsä useita resursseja siten, että ne estävät toinen toisensa toiminnan ja koko systeemi pysähtyy prosessien odottaessa toisiaan ristiin.

Kuvitellaan, että filosofit toimivat seuraavan algoritmin mukaan: 1) Ajattelee 2) Ota vasen haarukka 3) Ota oikea haarukka 4) Syö 5) Palauta haarukat alkuperäisille paikoilleen ja palaa kohtaan 1. Kaaviossa 3 on mallinnettu neljän näin toimivan filosofin muodostava systeemi Petri-verkkona. Ympyrät kuvastavat tiloja ja neliöt siirtymiä. Jokaista filosofia kohden on yksi ja jokaista haarukkaa kohden yksi merkki, joita on kuvassa merkitty mustilla palloilla. Kuvan tilanteessa kaikki filosofit ajattelevat ja kaikki neljä haarukkaa ovat pöydällä. V-kirjaimilla merkityt siirtymät vastaavat vasemman haarukan poimimista, o-kirjaimilla merkityt oikean haarukan poimimista ja p-kirjaimilla merkityt haarukoiden palauttamista.

Vasemman haarukan poimimisen virittämiseen tarvitaan yksi ajatteleva filosofi ja yksi haarukka-merkki, ja lauetessaan se johtaa tilanteeseen, jossa yhden ajattelevan filosofin ja haarukan merkit ovat kadonneet, mutta tilalle on tullut merkki paikkaan, joka kuvaa

filosofia, jolla on haarukka vasemmassa kädessä. Jos oikeanpuoleinen haarukka on käytettävissä, tämä virittää siirtymän, jossa oikeanpuoleinen haarukka katoaa ja vasemmassa kädessä haarukkaa pitelevä filosofi muuttuu syöväksi filosofiksi. Tämä virittää taas siirtymän, joka palauttaa molemmat haarukat pöydälle ja tuottaa ajattelevan filosofin.

Systeemi etenee siten, että joka askeleella yksi virittynyt siirtymä laukeaa. Jos samaan aikaan on useita virittyneitä siirtymiä, niin mielivaltaisesti valittu siirtymä laukeaa. Nyt voidaan formalisoida deadlockin käsite: deadlock on tilanne, jossa ei ole yhtään virittyneitä siirtymiä. Tämä voi tapahtua silloin, kun kaikilla filosofeilla on pelkkä vasen haarukka kädessä, jolloin kaikki filosofit näantyyvät nälkään. Se, onko systeemin mahdollista mennä deadlockiin, palautuu kysymykseen, onko deadlock-tila saavutettavissa alkutilanteesta.

Yleisemmin voi kysyä, onko systeemin mahdollista päätyä jostain alkutilasta johonkin toivottuun tai epätoivottuun tilanteeseen. Tämä niin sanottu saavutettavuusongelmia on todistettu hyvin vaikeaksi yleisessä tapauksessa, mutta sen ratkaiseminen tehokkaasti käytännön erikoistapauksissa on aktiivinen tutkimusalue. Yksinkertaisin tapa ratkaista ongelma pienille verkoille on koodata ohjelma, joka luettelee

tyhjentävästi verkon kaikki mahdolliset tilat, jotka ovat saavutettavissa alkutilanteesta. Filosofiverkolle tämä toimii, koska mahdollisten tilojen määrä on äärellinen ja pieni.

Deadlockin syntyminen tietokoneessa havaittiin käytännössä vasta sen jälkeen, kun ensimmäisiä suuria käytöjärjestelmiä oli kirjoitettu. Tarina liittyy Unix-käyttöjärjestelmän esihistoriaan. Nimi Unix on johdettu nimestä MULTICS (Multiplexed Information and Computing Service), joka oli ensimmäinen suuri tietokoneen käyttöjärjestelmä, joka oli kokonaan kirjoitettu korkean tason ohjelmointikielillä (PL/I-kieli).

Unix oli alunperin nimeltään Unics, ja nimi tulee todennäköisesti siitä, että sen ensimmäisten versioiden oli tarkoitus olla jonkin tyylinen ”yhden käyttäjän MULTICS”. Kun MULTICSia laadittiin, deadlockia ei osattu etukäteen ottaa huomioon. Se huomattiin vasta, kun MULTICS oli ensimmäistä kertaa deadlockissa.

Korkeamman tason Petri-verkot

Petri-verkkojen yksinkertaisuudesta ja rajoittuneisuudesta johtuen niillä voi olla vaikea mallintaa oikean maailman monimutkaisia systeemejä. Siksi verkkojen ilmaisuvoiman parantamiseksi on keksitty suuri määrä laajennoksia määritelmään. Yksi yleisesti käytetty laajennos on väritytty Petri-verkko, jossa merkeille annetaan jokin ”väri”.

Nyt merkit voivat vastata muuttujia tietokoneohjelmassa ja väri voi kertoa muuttujan arvon. Värien lisääminen merkeille ei kuitenkaan kasvata mallien laskentavoimaa teoriassa – minä tahansa väritytyn petri-verkon voi ilmaista myös monimutkaisempaan mutta yhtäpitävänä värittämättömänä verkkona. Värien käyttö voi kuitenkin helpottaa mallinnustyötä huomattavasti.

Nyky maailman IT:ssä termi ”värityt Petri-verkot” merkitsee työkalua, jossa on yhdistetty niin sanotusti perus-Petri-verkot sekä yksinkertainen mallinnuskieli, jolla voi rajoittaa verkon toimintaa erilaisilla lausekkeilla. Parhaiten dokumentoitu työkalu tähän lienee *cpntools*, jossa on paljon ominaisuuksia, mutta käyttöliittymän logiikka on omalaatuinen. Työkalulla on mahdollista määritellä ja piirtää verkkoja sekä analysoida niiden ominaisuuksia tietokoneavusteisesti.

Korkean tason Petri-verkoksi kutsutaan verkkoa, johon on liitetty tietyt lisäominaisuudet, jotka on määritelty ISO/IEC-standardissa. Tämä määritelmä on hyvin matemaattinen ja tekninen, mutta sen pääkohdat voidaan kuvata seuraavasti.

Korkean tason verkon pohjalla on tavallinen, paikoista ja siirtymistä koostuva Petri-verkko. Verkon paikoilla on nyt lisäksi tyyppi, hieman samaan tapaan kuin ohjelmointikielen muuttujilla voi olla tyyppi. Kaaret on merkitty kaavoilla, joiden suorittaminen tuottaa

joukon olioita. Siirtymät on merkitty totuusarvoisilla lausekkeilla, jotka voivat sisältää funktioita, jotka saavat argumentteinaan tulokaarista tulevia olioita. Siirtymä voi laueta silloin, kun siihen liitetty totuuslauseke on tosi, jolloin siitä lähtevät kaaret lähettävät lähtöpaikkoihin uusia olioita.

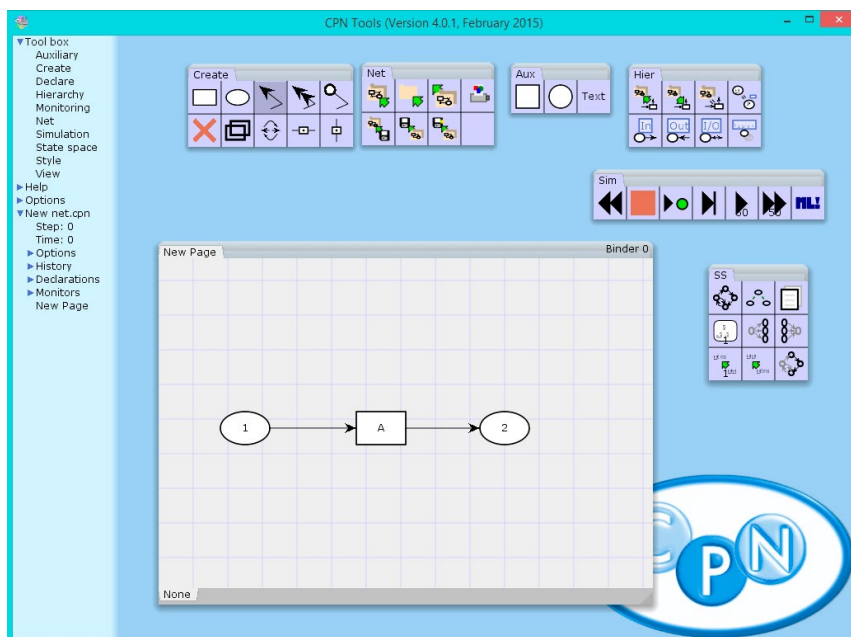
Korkeamman tason verkot vastaavat ilmaisuvoimaltaan yleiskäyttöistä ohjelmointikieltä, mikä tekee helpommaksi monimutkaisten järjestelmien mallintamisen. Huonona puolena on, että verkkojen analysoinnista voi tulla äärimmäisen vaikeaa tai jopa mahdotonta, koska verkko pystyy simuloimaan Turingin konetta, ja tietyt kysymykset Turingin koneen toiminnasta ovat todistetusti ratkeamattomia.

Hyödyllinen työkalu vai akateemista näpertelyä?

Vaikka Petri-verkkoja koskeva tutkimuskirjallisuus on laaja, on vaikea löytää konkreettisia menestystarinoita Petri-verkkoihin pohjaavalle mallinnukselle. Useiden suuryritysten kuten Boeing, Mitsubishi ja Siemens kerrotaan käyttävän Petri-verkkoihin pohjaavaa mallinnusta, mutta harvassa ovat tapaukset, joissa Petri-verkko on paljastanut järjestelmässä kriittisen ongelman, joka oli muuten jäänyt huomaamatta.

Ongelma on myös siinä, että mallinnuksen takana on ihminen, joten on mahdollista, että itse mallinnuksessa on virhe. Vaikka malli voitaisiin todistaa täysin virheettömäksi, on eri asia, vastaako malli oikeasti todellista järjestelmää. Kuinka voidaan varmistua, että malli on oikein?

Petri-verkkojen puolestapuhujat argumentoivat, että Petri-verkkoihin pohjaava mallinnus kannustaa formaalimpaan ajattelutapaan, jolloin muuten piilossa olevat ongelmat tulevat ilmiselviksi jo mallinnusvaiheessa. Petri-verkoilla mallinnettu järjestelmä lienee lopulta varmempi kuin ilman niitä suunniteltu järjestelmä. 🐞



Cpntools-työkalun käyttöliittymä.



Standardisodan taide

Sanonnan mukaan usein ihmetellään, miksi jollekin asialle on kaksi eri standardia. Ratkaisuksi ongelmaan kehitetään parempi standardi. Nyt standardeja on kolme.

Janne Sirén

Alakouluikäisenä erään luokkakaverini kotona oli aina 80-luvun kovimmat teknologiavermeet sekä kunnioitusta herättävä videokokoelma. Kun kaveri vielä oli mukava, niin viihdyin siellä usein ja videotkin pyörivät tiuhaan. Tajusin vasta myöhemmin – kun ystävän kotiin hankittiin toiset videot – että koko tuo videokokoelma oli VHS-nauhoja pienemmällä kaseteilla.

Sonyn kehittämän Betamaxin – lyhyemmin Betan – ja JVC:n lanseeraaman VHS:n kaksintaistelu lienee yksi teknologiamailman siteeratuimmista. Aina kun kaksi standardia kohtaavat, joku muistaa Betan hävinneen videosodan VHS:lle. Kaikilla on syykin huulilla: aikuisviihde. Se kel’ porno on, se standardisodan voittakoon.

Emme kaivanneet aikuisviihdeä kaverillani, mutta muistan Betassa olleen paremman kuvanlaadun. Ei ihme, että teknologisesti valvutuneemmassa kodissa sitä suosittiin. Mutta muistan myös Beta-nauhojen pienemmän kapasiteetin. Kun tv-ohjelmien tallentaminen oli tärkeää, pidempi tallennusaika oli VHS:lle etu.

Useita teknologiasotia sittemmin sivusta seuranneena en ole enää vakuutunut, että kumpikaan näistä puutteista oli syy Betan tappioon.

Olet vain niin hyvä...

Olin liian junnu nauhasotiin, mutta oman tulikasteeni sain, kun kamppai-

lu teräväpiirtovideon optisesta mediasta käynnistyi. Vastakkain olivat 2006–2008 Toshiba ja NECin kehittämä HD DVD sekä Sonyn, Philipsin ja Pioneerin Blu-ray. HD DVD sai puolelleen Microsoftin, joka vastasi standardin HDi-interaktiokerroksesta. Blu-ray sai varhaisiin tukijoihinsa yhdeksän suurta elektroniikkayhtiötä.

Suuri osa kaverini muinaisista Beta-nauhoista oli itse nauhoitettuja, mutta teräväpiirtolevyjen kohdalla avainroolissa olivat valmissisältöä julkaisevat suuret elokuvastudiot. HD DVD:llä oli Universalin tuki ja Blu-raylla Sonyn sekä 20th Century Foxin. Paramount ja Warner tukivat hetken molempia. Sodan lopettivat tuplaskut: Paramount siirtyi ensin HD DVD -leiriin ja Warner sitten Blu-rayhyn.

Jälkimmäinen ratkaisi sodan Blu-raylle, vaikka Blu-ray ei ollut mitenkään kiistatta johdossa. HD DVD ilmestyi ensimmäisenä, tarjosi aikaisemmin korkeatarkkuusaudiota ja sillä oli yksinoikeusstudioitakin. Blu-raylla oli suurempi kapasiteetti, mutta sen Java-pohjainen interaktiivisuus oli pitkään surkea verrattuna HD DVD:n sulavaan HDi-sisältöön. Silti HD DVD:n korttitalo romahti lopulta vuorokaudessa.

Toinen esimerkki löytyy sähköautoista. CHAdeMO on ensimmäinen laajalle levinnyt tasavirtapikalatausstandardi, jonka takana on kourallinen japanilaisia autoyhtiötä. Kun eurooppalaiset ja amerikkalaiset autofirmit havahtuivat

samaan tarpeeseen, ne lähtivät kehittämään kilpailevaa standardia – niinpä hieman myöhemmin ilmestyi CCS.

CCS kävi taistoon tosissaan. CHAdeMO:n onnistui vain vaivoin estää yritykset standardoida latausasemat CCS:ksi lakiteitse – kompromissina tiettyjen latausasemien EU:ssa tulee kuitenkin tarjota vähintään CCS-pistoke. Lailla standardeja on toki yritetty pakottaa aikaisemminkin: micro-USB:n standardointia matkapuhelien lataustavaksi ajettiin EU-direktiivillä, minkä Apple sai estettyä.

CHAdeMO ja CCS kamppailevat pikalatausstandardin asemasta tasapuolisesti, ja sähköautojen latausasemista löytyy yleensä molemmat. Kumpikaan standardi ei olennaisesti johda luvuissa, eikä kumpikaan juuri ole toistaan parempi. Kivenä kengässä suurin sähköautovalmistaja Tesla käyttää ihan muuta pikalatausliitintä sekä autoissaan että Supercharger-latausverkostossaan. Silti on enää ajan kysymys, että CCS tulee voittamaan sähköautojen maailmanlaajuisen pikalataussodan.

...kuin ystäväsi

Syy siihen, miksi VHS ja Blu-ray voittivat, ja miksi CCS tulee voittamaan, on yksinkertainen. Niiden takana on suurin osa markkinatoimijoista. Sony oli Betan kanssa pitkään yksin (virhe, jota se ei toistanut Blu-rayn kanssa) ja Toshiba HD DVD -kaverit olivat pieniä tai puolivillaisesti mukana. CHAdeMO ei ole yksinäinen puserrus, mutta se on yksinomaan japanilainen. Tasaväkinen varhaismenestys tai edes ensiksi ehtiminen eivät merkitse mitään. CHAdeMO:lta puuttuvat globaalit kumppanit. Kun sähköauto todella löytyi, vain yksi standardi tulee kasvamaan räjähdysmäisesti.

Poikkeus vahvistaa säännön: Applen ikiomat Lightning- ja sitä ennen Dockliittimet haastavat mobiili-USB:n ylivaltaa tasaväkisesti. Suoritusta pitää kunnioittaa. Hätäisimmät veikkaavat Teslan pystyvän samaan sähköautojen latauksessa, mutta ennustan, että Tesla murtuu. Applen suoritus on harvinaisen. Lähes aina muun markkinan joukkovoima voittaa, kun kriittinen massa saavutetaan.

Häviäjät voivat lohduttautua sillä, että voittajatkin joutuvat aikanaan väistymään seuraavan teknologiamurroksen tieltä. 🐘



Kuvat kuntoon

Paremmaksi valokuvaajaksi ohjelmistolla

Valokuvaus on helpompaa kuin koskaan. Lähes jokainen omistaa kameran, jos ei muuten niin osana matkapuhelinta. Automatiikan avulla kelvollisia kuvia syntyy vaivattomasti, mutta pienellä panostuksella kuvaamisesta saa vielä enemmän irti.

Teksti ja kuvat: Sakari Leppä, Mikko Rasa

Valokuvaus syntyi vuonna 1826, kun **Nicéphore Niépce** valotti näkymän ikkunastaan bitumipäällysteiselle metallilevyille. Tämä ensimmäinen valokuva oli erittäin epäselvä, mutta seuraavina vuosikymmeninä tekniikka kehittyi, ja vuosisadan loppuun mennessä filmi oli jo keksitty. Valokuvaus oli mullistava keksintö, sillä maisemat ja muotokuvat voitiin tallentaa silmänräpäyksessä esikuvallseen uskollisena. Perinteisellä taidemaalalla meni tunteja tai jopa päiviä maalaus tekemiseen, ja mallin piti olla liikkumatta maalaamisen ajan.

1900-luvun aikana optiikka ja filmit kehittyivät. Keksittiin zoom-objektiivit ja väri filmi. Vuosisadan puolivälissä Polaroid toi markkinoille kameran, joka tuotti välittömästi käyttökelpoisia paperikuvia. Pikakamera saavutti helppokäyttöisyytensä ansiosta suurta suosiota, mutta erikseen kehitettävä filmi säilytti asemansa vakavampien kuvaajien keskuudessa.

Seuraava suuri kehitysaskel oli elektroninen kuvasensori. Tekniikkaa kehitettiin jo 1970-luvulla, mutta kuluttajamarkkinoille ensimmäiset laitteet saapuivat vasta 1980-luvun lopulla.

Vuodesta 2003 alkaen digitaalikameroita on myyty enemmän kuin filmikameroita. 2000-luvun aikana myös kamerapuhelimet ovat löytäneet tiensä lähes jokaisen taskuun.

Digitaalikameran toiminta

Digitaalikameroita on karkeasti kolmea tyyppiä: kamerapuhelimia, kompaktikameroita ja digitaalisia järjestelmäkameroita. Kuvanlaatuun vaikuttaa olennaisesti optisten komponenttien kyky kerätä valoa, joten paremmat kamerat ovat myös kooltaan isompia. Järjestelmäkamera objektiivineen voi kuitenkin painaa monta kiloa, ja sen kuljettaminen on vaivalloista. Monesti paras kamera onkin se, joka on mukana.

Kuten perinteisessä filmikamerassa, myös digitaalikamerassa kohteesta tuleva valo kulkee objektiivin linssien läpi ja muodostaa kuvan kuvapinnalle. Tähän yhtäläisyydet loppuvatkin. Filmiin osuva valo aiheuttaa filmin emulsiassa kemiallisen reaktion, kun taas digitaalikameran kuvakenno muuntaa siihen osuvat fotonit sähköisiksi signaaleiksi. Kuvakennon fotonit-ilmalaisimet eivät välitä fotonien aallonpituudesta, joten värikameroissa kuvakennon edessä on

suodin, joka päästää tietyt aallonpituudet vain tiettyihin pikseleihin. Lähes kaikissa kameroissa käytetään Bayer-suodinta, jossa neljän pikselin ryhmä koostuu kahdesta vihreästä, yhdestä punaisesta ja yhdestä sinisestä pikselistä.

Kuvakennolta saatavat fotonilukemat eivät vielä sellaisenaan ole valmis valokuva. Kameran ohjelmisto käsittelee kennon kaappaamaa dataa tuottaakseen katsottavaksi sopivan kuvan. Yleensä näiden toimintojen voimakkuuteen pystyy vaikuttamaan, mutta vain rajallisesti. Esimerkiksi kameran kohinanpoisto saattaa olla liian voimakas tai muuten huonolaatuinen.

Monissa kameroissa on tarjolla myös mahdollisuus ottaa raakakuvia. Aluksi tämä oli vain järjestelmäkameroiden ominaisuus, mutta sittemmin se on yleistynyt myös kompaktikameroissa ja löytyy nykyään jopa joistakin kamerapuhelimista. Raakakuvaan tallennetaan sensorin antamat lukemat sellaisenaan, ilman käsittelyä. Bittisyvyys on tyypillisesti 12–16 bittiä, ja kuva tallennetaan häviöttömällä pakkauksella. Näin tallennettu kuva antaa jälkikäsitteilyyn paljon enemmän pelivaraa kuin valmiiksi prosessoitu JPEG-kuva.



Canon EOS 5D Mark II, digitaalinen järjestelmäkamera (2008)



Canon EOS M, digitaalinen peilitön järjestelmäkamera (2012)

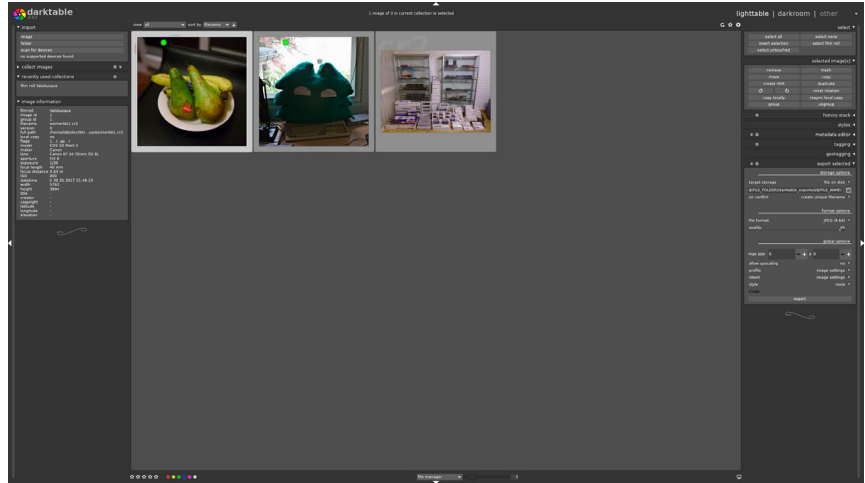


Canon PowerShot A40, digitaalinen kompaktkamera (2002)

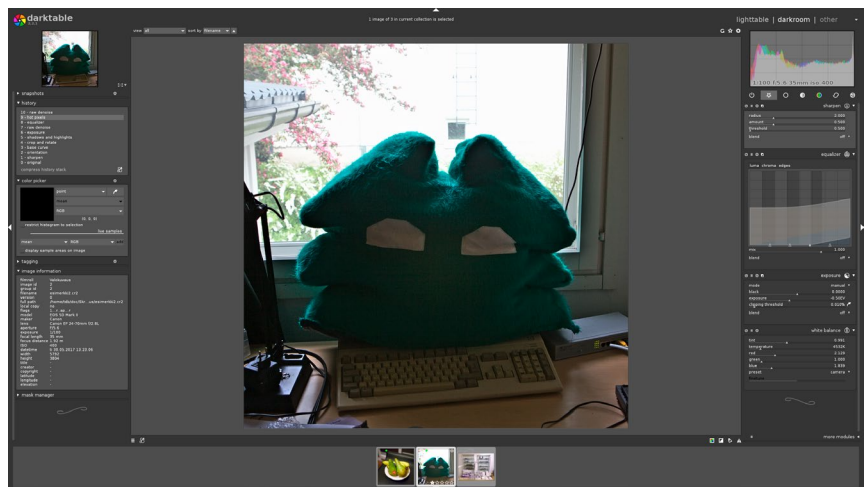
Vinkkejä kuvien ottamiseen

Aivan kaikkea ei voi korjata jälkikäsitelyssä. Kameran automatiikka tuottaa tavallisimmissa tilanteissa tyydyttävän lopputuloksen, mutta opettelemalla valokuvauksen perusteet voi laatua parantaa etenkin hämärässä ja muuten hankalissa olosuhteissa.

Valitse kuvausparametrit tilanteen mukaan. Jos kuvaat liikkumatonta kohdetta ja voit yrittää otosta useita kertoja, valitse aluksi parametrit, jotka antavat parhaan kuvanlaadun. Yleensä tämä tarkoittaa matalaa herkkyyttä ja siten pitkää valotusaikaa. Jos kuva ei ota onnistuakseen, voit löysentää parametreja ja yrittää uudestaan. Arkoja kohteita tai nopeasti muuttuvia tilanteita kuvatessa on parasta käyttää lyhyttä valotusaikaa ja pientä aukkoa, jotta yksittäinenkin kuva todennäköi-



Valopöytänäkömä tarjoaa työkalut kuvakokoelman järjestelyyn.



Pimiönäkömässä kuvaa voi käsitellä.

semmin onnistuu.

Tarkennukseenkin on syytä kiinnittää huomiota. Kameran tarkennusautomaattikka osuu yleensä kohdalleen, kunhan valitset tarkennuspisteen oikein. Huomioi myös aukon vaikutus syväterävyyalueeseen. Aukko onkin usein tärkein kuvausparametri, jonka ehdoilla muut valitaan.

Koska digitaaliset valokuvat eivät maksa juuri mitään, kuvia kannattaa ottaa paljon. Kun otat samasta kohteesta pari-kolme kuvaa peräjälkeen, kasvatat mahdollisuuksia sille, että ainakin yksi niistä onnistuu.

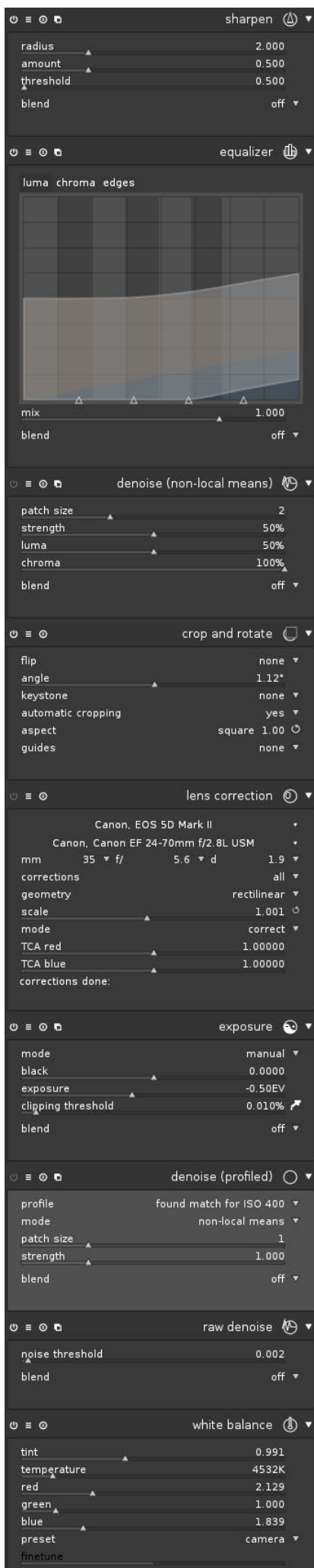
Valokuvien käsittely

Kun muistikortti on sitten kuvattu täyteen raakakuvia, ne täytyy käsitellä katseltavaan muotoon. Jonkin verran käsittelyä voi tehdä tavallisella kuvankäsittelyohjelmalla kuten GIMPillä tai Photoshopilla, mutta pidemmän päälle samojen operaatioiden toistaminen käy työlääksi. Siksi puuhaan onkin tehty erilliset ohjelmat. Suosituin kauppal-

inen ohjelma on Adobe Lightroom, mutta tässä artikkelissa käytämme avoimen lähdekoodin Darktablea. Darktable löytyy useimmista Linux-jakeluista, ja sen saa binääriin myös MacOS:lle, mutta Windows-käyttäjät joutuvat toistaiseksi kääntämään omansa lähdekoodista. Useimmat mainitut asiat ovat sovellettavissa myös Lightroomiin.

Darktablen käynnistyessä esillä on valopöytänäkömä (lighttable). Suurimman osan näytöstä täyttää kuvien selailuun käytettävä alue, joka on vielä tyhjä. Vasemmasta laidasta löytyvät työkalut kuvien tuomiseksi ohjelmaan (import). Voit tuoda joko yksittäisiä kuvia (image) tai kokonaisen hakemiston kerralla (folder). Oikeassa laidassa on kuvien järjestelyyn liittyviä toimintoja.

Tuotuaasi kuvat voit siirtyä pimiönäkymään (darkroom) kaksoisnapsauttamalla kuvaa. Nyt valittu kuva täyttää ruudun keskialueen, ja alareunaan ilmestyy kuvapalkki, josta voit siirtyä



Joitakin yleisimmin käytettyjä Darktablen toimintoja.

kokoelman muihin kuviin. Vasemmassa reunassa näkyy käsiteltävänä olevan kuvan tiedot ja oikealla mahdolliset käsittelytoiminnot. Darktable poikkeaa perinteisestä kuvankäsittelyohjelmasta sikäli, että alkuperäiseen kuvatiedostoon ei kosketa. Valitut käsittelytoiminnot suoritetaan lennossa uudestaan joka kerta, kun parametreja muutetaan, ja tulos pidetään muistissa. Parametrit tallennetaan kuvatiedoston rinnalle erilliseen tiedostoon.

Käsittelytoiminnot on ryhmitelty viiteen kategoriaan: perustoiminnot (basic), sävyt (tone), värit (color), korjaukset (correction) ja tehosteet (effects). Näiden lisäksi on kaksi meta-kategoriaa omille suosikkitoiminnoille ja nykyisessä kuvassa aktiivisille toimintoille. Toimintojen suoritusjärjestys on ennalta määrätty. Seuraavaksi käydään läpi joitakin yleisimpiä toimintoja.

Kuvan perushuolto

Valotuksen säätöä (exposure) voi käyttää ali- tai ylivalottuneen kuvan korjaamiseen. Se toimii kuten herkkyyden säätö kamerassa – valoisuuden lisääminen tuo mukanaan kohinaa. Mittayksikkönä on EV (exposure value), joka on logaritminen ja suhteellinen asteikko. Yhden EV:n muutos tarkoittaa valomäärän kaksinkertaistumista tai puolittumista. Tummaa kuvaa voi yleensä kirkastaa pari–kolme EV:tä laadun säilyessä siedettävänä, mutta yli yhden EV:n ylivalotuksen korjaaminen voi olla vaikeaa.

Oikean valotuksen ohella toinen kuvan luonnollisuudelle tärkeä tekijä on valkotasapaino (white balance). Kamera tallentaa raakakuvan metatietoihin oman arvauksensa, joka kelpaa vähintäänkin lähtökohdaksi. Darktablen valkotasapainotoiminnolle kerrotaan kuvaustilanteesta vallitseva värilämpötila, ja ohjelma korjaa sen perusteella kuvan sävyt neutraaleiksi. Myös kuvan vihertävyyttä voi säätää, mutta useimmiten se ei ole tarpeen.

Hyvälläkin kameralla kuvaan päätyy aina hieman kohinaa. Sen poistamiseen (denoise) on tarjolla useita eri toimintoja, jotka toimivat hieman eri tavoin. Kääntöpuolena kaikissa on kuvan yksityiskohtien pehmentyminen.

Raakapikselien kohinanpoisto (raw denoise) suoritetaan ennen raakakuvan Bayer-mosaikin muuntamista

RGB-arvoiksi. Se on erittäin nopea ja tuottaa pienillä kohinamäärillä hyvän lopputuloksen, mutta ei kykene erottelamaan kirkkaus- ja värikohinaa.

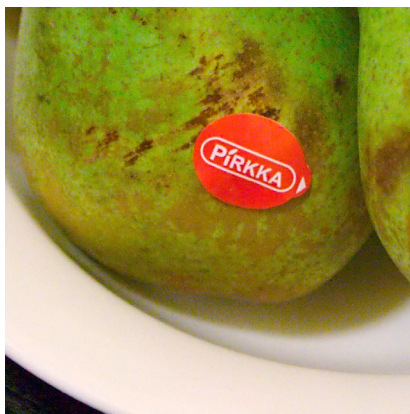
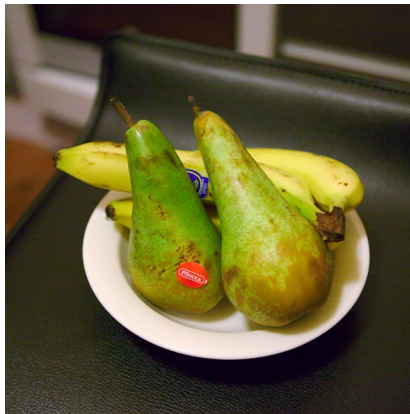
Epäpaikallisiin keskiarvoihin perustuva kohinanpoisto (denoise (non-local means)) käyttää raskaampaa algoritmia mutta tuottaa paremman lopputuloksen. Toiminnon voimakkuutta voi säätää erikseen kirkkaus- ja värikohinalle. Tästä on hyötyä voimakkaasti kohisevilla kuvilla, koska värikohina on häiritsevämpää ja liiallinen kirkkauskohinan poistaminen tekee kuvasta lattean. Algoritmi tosin on raskaampi, ja laskenta kestää pidempään.

Profiloitu kohinanpoisto (denoise (profiled)) sisältää valmiiksi määritellyt parametrit lukuisille eri kameramalleille ja herkkyyсарvoille. Sisäisesti toiminto käyttää joko epäpaikallisia keskiarvoja tai aallokemuunnosta. Eri toimintoja voi myös käyttää yhdessä, hyödyntäen kunkin vahvuuksia.

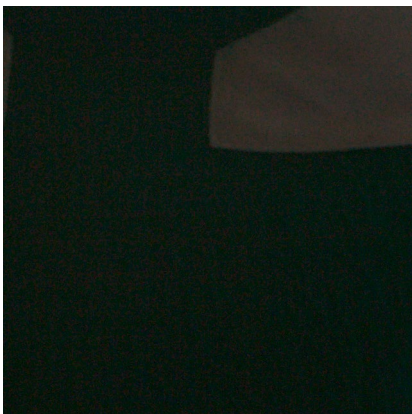
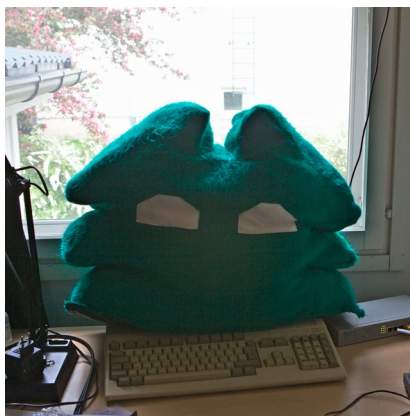
Optiikkakaan ei ole täydellistä. Kaidissa objektiiveissa on jonkin verran vinjetointia (vignetting), eli kuvan nurkat ovat keskiosaa tummempia. Etenkin zoom-objektiiveissa esiintyy myös tynnyri- tai tynnyvääritystä, jossa suorat viivat kuvan laidoina kaa-reutuvat sisään- tai ulospäin. Nämä optiset virheet voidaan kompensoida lähes täydellisesti matemaattisen mallin perusteella, jos objektiivin ominaisuudet tunnetaan. Darktablen linssikorjaus-toiminnossa (lens correction) on tietokanta yleisistä objektiiveista, ja usein korjaus onnistuu automaattisesti nappia painamalla.

Terävöitys (sharpen) on olennainen toiminto, koska kuvakennolta tuleva pikselidata on aina hiukan pehmeää. Darktable kytkeekin terävöityksen päälle ilman eri pyyntöä. Oletusasetus on oikein tarkennetulle kuvalle sopiva. Yliterävöitystä kannattaa välttää, koska se aiheuttaa esineiden ympärille häiritseviä sädekehjiä.

Darktablen ehkä monipuolisin käsittelytoiminto on equalizer. Se on aallokemuunnokseen (wavelet transform) perustuva käsittelytoiminto, jolta onnistuu niin kohinanpoisto ja terävöitys kuin kontrastin ja värikkyyden säätö. Parametreja voi säätää eri kokoisille kuvan rakenteille, joten samanaikaisesti voi vaikkapa tasata kuvan eri alueiden kirkkauseroja ja tuoda esiin pieniä yksityiskohtia.



Esimerkki 1. Ilman salamaa otettu kuva on valotuksen korjaamisen jälkeen kellertävä. Se korjaantuu valkotasapainon korjauksella. Lopuksi poistetaan kohina.



Esimerkki 2. Jos vastavaloon kuvaamista ei voi välttää, Shadows & Highlights -toiminto auttaa. Myös kokonaisvalotusta on tarpeen laskea hieman. Viimeistelet poistamalla kohina.

Equalizerin käyttöliittymän pääkomponentti on käyränäkymä. Näkymän vasen reuna vastaa kuvan laajoja alueita, oikea reuna pieniä yksityiskohdita. Käyrän pisteitä voi raahata hiirellä pystysuunnassa. Neutraalin keskikohdan yläpuolella korostetaan kyseisen kokoisia piirteitä, alapuolella vaimennetaan niitä. Näkymän alareunassa on toinen, kohinatasoon vaikuttava käyrä.

Käyränäkymän yläpuolella on kolme painiketta: valoisuus (luma), värit (chroma) ja reunat (edges). Kullakin näistä on omat käyränsä. Ihmisen näköjärjestelmä havaitsee kirkkauden ja värisävyjen vaihteluita eri tavoin, joten näille on usein järkevää käyttää erilaisia asetuksia.

Jos todellisuukselle uskollinen kuva ei miellytä, siitä voi tehdä mielenkiintoisemman tehosteilla. Tarjolla on laaja valikoima erilaisia taiteellisia toimintoja, joista suurin osa perustuu värien muokkaamiseen eri tavoin. Myös joillekin korjaustoiminnoille löytyy vaihtoehtoisia käyttötarkoituksia tehosteina. Etenkin equalizerilla saa aikaan todella näyttäviä tuloksia.

Kourallisen kuvia käsiteltyään alkaa saada tuntuman siitä, mitkä asetukset sopivat omalle kameralle ja miellyttävät omaa silmää. Nämä asetukset voi tallentaa esivalinnoiksi, jolloin niitä on helppo käyttää jatkossa. Monissa toiminnossa tulee valmiina joukko esivalintoja, joista voi hakea vinkkejä toimintojen käyttöön.

Käytännön esimerkkejä

Katsokaamme lopuksi vielä muutama käytännön esimerkki. Esimerkki 1 on pahasti alivalottunut. Näin voi käydä hämärissä olosuhteissa ilman salamaa, tai jos salama ei välähdy toivotusti esimerkiksi paristojen loppumisen takia.

Aloitamme korjaamalla valotuksen. Esimerkkikuvalle sopiva arvo on +2 EV. Seuraavaksi huomaamme kuvan olevan kellertävä, koska se on otettu lämminsävyisessä keinovalossa. Valkotasapainon korjauksessa auttaa ikkunan oikeassa ylänurkassa oleva histogrammi. Väriämpötila on laskettava noin 2000 kelviniin.

Koska nostimme valotusta merkittävästi, kuva on nyt huomattavan kohinainen. Käytämme kohinanpoistoon raw denoise- ja equalizer-toimintojen yhdistelmää. Raw denoisen kynnyksarvoksi sopii 0,04. Darktablen mukana

toimitettava equalizerin esivalinta ”denoise & sharpen” on sopiva lähtökohta, mutta sellaisenaan turhan voimakas. Niinpä teemme siitä kevennetyn version.

Esimerkissä 2 Skrolli-trolli istuu tieteknisen klassikon päällä. Pahaksi onneksi trollin takana olevasta ikkunasta avautuu näkymä kirkaaseen ulkoilmaan ja trolli istuiminen on alivalottunut. Vastavaloon kuvaamista kannattaa pyrkiä välttämään, mutta joskus se ei onnistu. Tässä tilanteessa emme voi yksinkertaisesti nostaa koko kuvan valotusta, koska se polttaisi ikkunasta näkyvän maiseman sekä trollin yläosat puhki.

Hätiin tulee varjot ja korostukset -toiminto (shadows and highlights). Aivan ääripäihin se ei pure, joten ikkunan tummentamiseksi laskemme myös kokonaisvalotusta hieman. Sopivat arvot ovat valotus -0,5 EV, varjojen korjaus +75 ja korostusten korjaus -30. Kohinanpoisto onnistuu samoin kuin edellisessä kuvassa.

Esimerkissä 3 on Skrollissa 2016.1 julkaistu pienoisrautatietekokoelman kuva, jonka käsittelystä sain idean tämän jutun tekemiseen. Kuva on otettu pimeänä talvi-iltana, pienellä aukolla ja peräti kuuden sekunnin valotusajalla. Ensivaikutelma on varsin epäonnistunut: keinovalosta johtuen kuva on kauttaaltaan keltainen ja sen oikea laita on huomattavasti vasenta vaaleampi. Kuvan reunojen lähellä suorista linjoistakin on tullut kaarevia objektiivin vääristymien takia. Vaan eipä masennuta vielä, vaan katsotaan, mitä käsittelyllä saa aikaan.

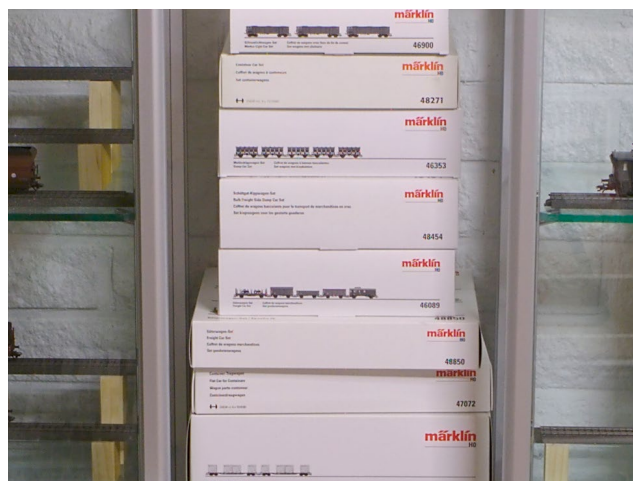
Kytkeämme aluksi linssikorjaustoiminnon päälle. Se suoristaa kuvan linjat ja kirkastaa nurkkia hieman. Valaistuksen epätasaisuuden voimme kompensoida käyttämällä liukusävytettyä harmaasuodinta (graduated density). 2 EV:n suodin -90 asteen kulmassa tuottaa silmää miellyttävän lopputuloksen. Seuraavaksi korjaamme värilämpötilaksi noin 2250 kelviniä. Kuva jää hieman tummaksi, joten nos-

tamme valotusta 0,7 EV:n verran.

Lopuksi poistamme kohinan ja suoristamme kuvan. Vinoutta ei ole kuin puoli astetta, mutta tiiliseinä saumasta se erottuu selvästi. Lopputulos on kuin paraskin studiokuva.

Tässä esitetyt kuvat asetuksineen voi ladata Skrollin sivuilta osoitteesta skrolli.fi/numerot.

Darktablen käyttöoppaasta löytyy ohjeita ja esimerkkejä myös muille kuin tässä esitellyille toiminnoille: www.darktable.org/usermanual/index.html.php 🐱



Esimerkki 3. Pimeänä talvi-iltana otettu kuva kärsii myös linssivääristymästä. Valaistuksen epätasaisuutta voi säätää harmaasuotimella.

Virtuaalirintamalta jotain uutta

Skrolli on seurannut Samsung/Oculuksen ja Googlen johtamien virtuaalitodellisuusliittoumien irrallista, mutta samansuuntaista kehitystä (ks. 2016.2 ja 2017.1). Vielä lienee ennenaikaista julistaa mobiilikeinotodellisuuden alustasotaa päättyneeksi (saati kunnolla edes alkaneeksi), mutta tervetullutta lähentymistä on tapahtumassa Samsungin Gear VR - ja Googlen Daydream-alustojen välillä.

Ensinnäkin Samsung on keväällä tuonut Gear VR -alustalle Daydream-ohjaimen (kuvassa yllä) kaltaisen Gear VR Controllerin (alla), mikä on tervetullutta molempia alustoja koskeville kehittäjille. Pelit ja ohjelmat on nyt helpompi sovittaa molemmille. Gear VR:n ohjain on pääosin samanlainen kuin Googlen ohjain, vain hieman ergonomisempi. Lisäksi alla oleva ylimääräinen liipasin tekee ampumisesta ja tarttumisesta helpompaa.

Skrolli kokeili Samsungin uusittua Gear VR -silmiä liisäohjaimella sekä Samsung Galaxy S8:aa. Laseja on muilta osin tuunattu vain vähän ja niitä voi käyttää entiseen tapaan ilman ohjaintakin. Gear VR -ilotikun voi myös ostaa erikseen vanhemmille silmiöille. Vaikka S8:n pikselitiheys ei ole olennaisesti muuttunut, leveämpi wqhd+ -näyttö laventaa näkökenttää ja lisätehot auttavat. Edelleen parasta mobiili-vr:ää.

Toisekseen Daydream-tuki on tulossa tänä kesänä Samsungin Galaxy S8 -puhelinperheelle. Rauta ylittää jo vaatimukset, mutta ohjelmistopuolella valmistajien on vielä tanssittava tanssinsa. Tarvittavien silmiöiden määrää uutinen ei valitettavasti tule vähentämään – Daydream vaatii edelleen omat lasinsa. 🐞

Janne Sirén

SD-512 – MSX muistikorttiaikaan

Vanhon tietokoneiden harrastamisen suurin este on usein sisällön siirtäminen ullakolta tai verkkokirppikseltä löytyneeseen koneeseen. Kotimikrot käyttivät levykkeitä tai kasetteja, jotka tuppaavat nykymittapuulla olemaan hitaita, hankalia hankkia ja yhä epäluotettavampia. Ratkaisuna 1980-luvun tietokoneisiin onkin saatavilla kattava joukko uudempia massamuistiratkaisuja, joiden hinta vaihtelee muutamasta kympestä useampaan sataseen.

Kerroimme hiljattain ratkaisuista Commodore 64:lle (katso Skrolli 2017.1), ja nyt vuorossa on Microsoftin kotitietokonealusta MSX. SD-512 on MSX/MSX2/MSX2+/Turbo-R-yhteensopiva laajennus, jolla MSX-standardin mukaiset koneet voivat käyttää ohjelmia SD- ja microSD-muistikorteilta. SD-512 liitetään MSX-koneen moduuliporttiin. Siinä on kaksi korttipaikkaa (yksi kumpaakin kokoa) sekä 512 kilotavun muisti, joka toimii tarvittaessa myös MSX:n lisämuistina.

Valmistelut tehdään 2000-luvun koneella: SD-muistikortti alustetaan käyttämään FAT32-tiedostojärjestelmää ja sinne puretaan SD-512:n ohjelmisto. Sitten MSX-pelit ja -ohjelmat tallennetaan kansioihin muistikortille. Moduulin kytkemisen jälkeen MSX käynnistyy suoraan tiedostoselaimen, josta valitaan haluttu ohjelma peliohjaimella.

SD-512:ta myy 8bits4ever.net 70 euron hintaan. Sisartuotteilla CF-640, ATA-620 ja FDC-600 myös Compact Flash -muistikortit, kiintolevyt ja pc-levykeasemat kytkeytyvät MSX:ään. 🐞

Ismo Utriainen





Vattukoneella lentoja seuraamaan

Raspberry Pi -pikketietokone, muutama ohjelmisto ja kourallinen edullisia lisäosia loivat ruudulle kartan lähistöllä liikkuvista lentokoneista.

Teksti: Jarkko Lehti Kuvat: Kimmo Rinta-Pollari, Jarkko Lehti

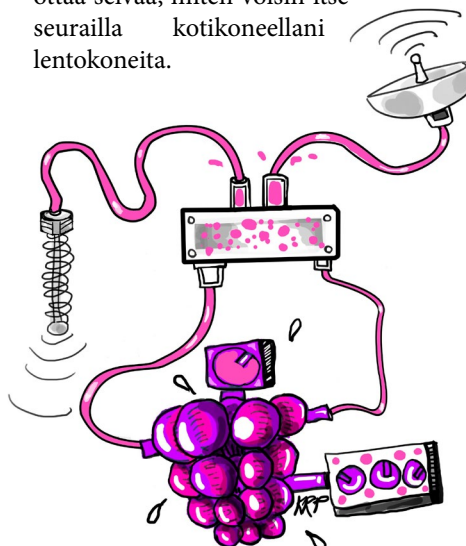
Aika monella meistä on Raspberry Pi hankittuna kovan innostuksen saattamana, mutta varmasti monella, niin kuin itselläkin se kuitenkin jäi pölyttymään pöytälaatikkoon. Pisteenä i:n päälle niitä on tullut hankittua lisää aina uuden version ilmestyessä, arkipäivän Sulo Vilénin tavoin ohittamattoman tarjouksen noustessa esiin.

Olen jo pienestä asti ollut kiinnostunut radiosta. Muistan lapsena kuunnelleeni pimeinä talvi-iltoina vanhasta putkiradiosta lyhytaaltoalueita. Lukuisten puhe- ja musiikkikanavien lisäksi sieltä kuului monenlaista piipitystä ja pulputusta. Myöhemmin minulle selvisi, että kyseessä oli datalähetyksiä, mutta mitään konstia niiden dekodointiin minulla ei aikanaan ollut. Vuodet vierivät ja tietotekniikan tarjoamat mahdollisuudet paranivat, joskin myös lyhytaaltoalueet hiljenivät niin, ettei niiltä nykyään kuule juuri muuta kuin kiinalaisia ja venäläisiä lähetyksiä, pulputuksesta puhumattaakaan. Niinpä innostuskin hiipui pitkäksi aikaa.

Jokunen vuosi sitten alkoi netissä kuulua kovasti hypeä softaradioista, missä radiolähetykset otetaan tietokoneelle vastaan raakana ja ohjelmallisesti sitten dekodataan sieltä halutussa formaatissa lähetykset ulos. Tähän innostuksen lähde oli niin sanottu RTL-SDR-ajuri, jolla saadaan halvoista Realtekin 2832U-piirin sisältävistä

USB-väylään liitettävistä digi-TV- ja FM-radiovastaanottimista raakadata pihalle ohjelmallista dekodointia varten. Tyypillisesti noin 20 dollarin laitteella saatiin tehtyä sama kuin huomattavasti kalliimmilla ammattilaitteilla.

Olen myös aina katsellut taivaalla lentäviä lentokoneita mielenkiinnolla ja miettinyt, mihin mikin kone on matkalla. Tähän tarjosi vastauksen netin Flightradar-palvelu, jossa lentokoneita pystyy seuraamaan kartalla ja niitä klikkaamalla saamaan tarkempia tietoja lennoista. Aloin kuitenkin ihmetellä, mistä tämä data tulee, ja yllätyin kovasti kun tajusin, että data tulee ihmisten kotikoneilta, jotka kuuntelevat lentoliikennettä ja välittävät sen palveluun prosessointia varten. Tämä alkoi kiehtoa niin paljon, että päätin ottaa selvää, miten voisin itse seuralla kotikoneellani lentokoneita.



Hommattuani Ebaysta soveltuvan RTL-SDR-vastaanottimen mietin, että tähän käyttöönhan on järkevintä valjastaa pöytälaatikossa yksinäisyyttään itkenyt Raspberry Pi 2.

Vattututkan asennukseen

Ensimmäiseksi Raspberry Pi -koneen Raspbian-käyttöjärjestelmälle asennetaan toimiva ympäristö. Lähtökohta on tuore sd-kortille kirjoitettu Raspbian ja ensimmäinen käynnistys. Tätä samaa prosessia voi käyttää aika pitkälle minkä tahansa muunkin Linux-jakelun kanssa olettaen, että jakelu tarjoaa tarvittavat paketit.

Kirjautu sisään komentoriviltä. Käyttäjätunnus on **pi** ja salasana **raspberry**.

Seuraavaksi päivitämme käyttöjärjestelmän ja Raspberry Pi:n firmwaren ajan tasalle syöttämällä seuraavat komennot:

Vastaa myöntävästi kysymyksiin. Kun salasanaa kysytään, kirjoita sama kuin edellä.

```
sudo apt update
sudo apt upgrade
sudo apt install rpi-update
sudo rpi-update
sudo reboot
```

Bootin jälkeen kirjautu sisään ja jatka komennolla

```
sudo apt install rtl-sdr
```

Kytke RTL-SDR-vastaanotin USB-liittimeen ja kirjoita:

```
rtl_test
```

Seuraavaksi järjestelmä ilmoittaa löytäneensä soveltuvan vastaanottimen ja kertoo siitä tietoja:

```
Found 1 device(s):
0: Realtek, RTL2838UHIDIR, SN:
00000060
Using device 0: Generic RTL2832U
OEM
Found Elonics E4000 tuner
Supported gain values (14): -1.0
1.5 4.0 6.5 9.0 11.5 14.0 16.5
19.0 21.5 24.0 29.0 34.0 42.0
Sampling at 2048000 S/s.
```

Seuraavaksi asennetaan varsinaisen ohjelman lentokoneiden seuraamista varten. Se pitää asentaa lähdekoodista, joten jatka näillä komennoilla:

```
sudo apt install git
git clone git://github.com/
MalcolmRobb/dump1090.git
cd dump1090
sudo apt install librtlsdr-dev
sudo apt install libusb-1.0-0-dev
make
```

Jos kaikki meni niin kuin Strömössä, meillä on nyt käännettynä dump1090-sovellus joka vastaanottaa dataa 1090 megahertsin taajuudella, dekodaa sen ja visualisoi havaitut lentokoneet kartalla. Saamme ohjelman käyntiin kirjoittamalla

```
./dump1090 --interactive --net
--net-http-port 8080
```

Ota nettiselaimella yhteys Raspberry Pi'n IP-osoitteeseen porttiin 8080. Näin voit katsella karttanäkymää, jossa lentokoneiden pitäisi näkyä – olettaen, että antenni saa vastaanotettua signaaleja.

Osoite on yleensä lähiverkossa <http://raspberrypi:8080/>, jos reititin osaa määritellä nimet paikallisverkon osoitteille. Jos ei, sinun pitää käyttää ip-osoitetta, esimerkiksi Raspberry Pi-koneen oman graafisen käyttöliittymän kautta <http://127.0.0.1:8080/>

Antenneista

Digi-TV-vastaanottimen mukana toimitetaan tyypillisesti jonkinlainen piiska-antenni, joka ei ole kovinkaan hyvä, mutta myös sillä on mahdollista vastaanottaa 1090 megahertsin taajuusalueella. Antennia voi hieman parantaa virittämällä se oikealle taajuusalueelle katkaisemalla se oikean mittaiseksi.

Ymmärtääksesi paremmin anten-



Kotitekoisella antennilla voi parantaa signaalien löytymistä.

nia tai pikemminkin radioaaltoja voit mielessäsi visualisoida, miten antenni lähettää maailmalle valonnopeudella etenevää ”naru”, joka aaltoilee sini-aaltona. Koska lentokoneen lähten sykkii 1090 megahertsin taajuudella eli 1090 miljoonaa kertaa sekunnissa ylös ja alas, voit kuvitella, miten lähettimen antennista lähtevä naru tekee samaa. Signaalin aallonpituus voidaan laskea jakamalla valonnopeus metreinä sekunnissa signaalin taajuudella. Tässä tapauksessa: $299\,792\,458\text{ m/s} / 1090\,000\,000\text{ Hz} \approx 0,275\text{ m}$, eli yhden aallon pituus on 27,5 cm.

Koska vastaanottimen mukana tullut antenni on lyhyempi kuin tämä, järkevintä on käyttää $\frac{1}{4}$ aallon mittaista antennia, jolloin pituudeksi saadaan noin 6,9 cm. Katkaisemalla mukana toimitettu antenni 6,9 sentin mittaiseksi siitä saa siis teoriassa paremman. Vastaanottopäässä ei kuitenkaan ole ihan niin tarkkaa antennin oikea mitoitus kuin lähetyksessä.

Toinen vaihtoehto on rakentaa itse soveltuva antenni, jolloin ainakin itse olen päässyt huomattavasti parempiin tuloksiin. Oheisen kuvan mukainen antenni on koottu juottamalla liittinrunkoon kahden millimetrin kupariputkea. Sitä voisi vielä parantaa lisäämällä neljä maatasopiikkiä liittimen alaosaan. Parhaiten vastaanottoa parantaisi antennin vieminen ulos talon katolle, jolloin se pitäisi koteloida säänkestäväksi. Kotelointia tehdessä tulee ottaa huomioon, ettei kotelon materiaali saa vastustaa radiosäteilyä. Tämä on helppo testata mikroaaltouunissa: jos kotelo lämpenee mikroaaltouunissa, se ei sovellu sääsuojaksi.

Internetistä löytyy runsaasti aiheesta informaatiota. Tietoa löytyy esimerkiksi siitä, miten voit itse ottaa osaa Flightradar-palveluun lähettämällä omat havaintosi koneista. Samaten on tietoa parempien antennien rakentamisesta ja jopa siitä, miten itse vastaanotinta voi muokata paremmaksi. 🛩️

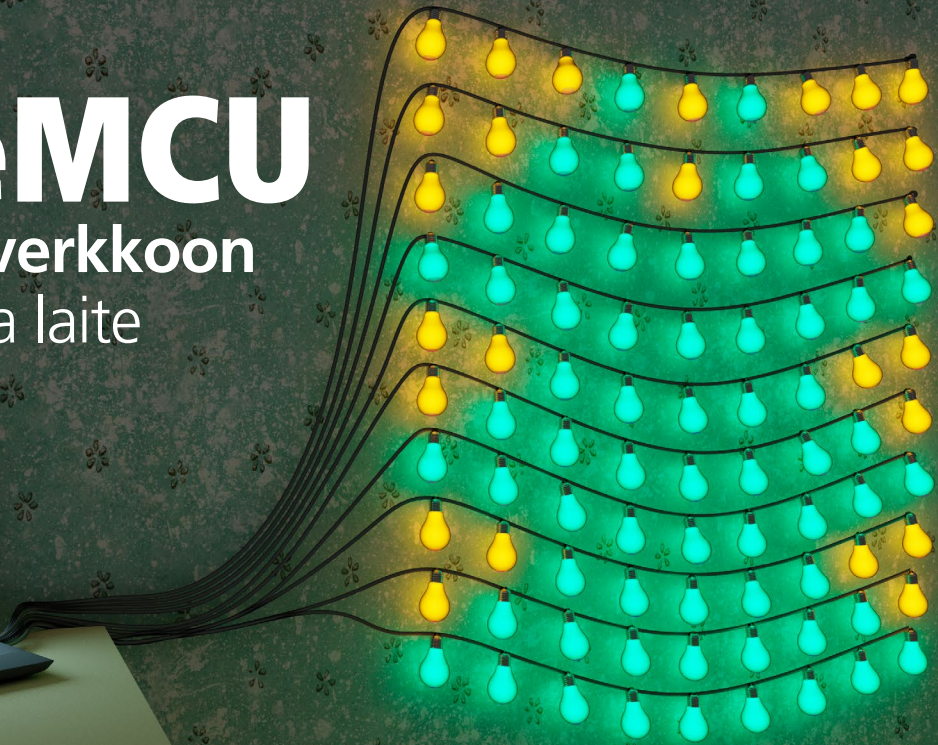
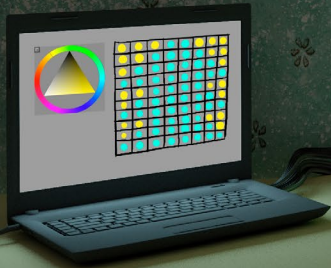
Linkkejä

- osmocom.org/projects/sdr/wiki/Rtl-sdr
- github.com/MalcolmRobb/dump1090
- www.flightradar24.com/

NodeMCU

avain esineverkkoon

Rakenna oma laite ESP8266:sta



Innostuitko esineiden internetistä? Viime numerossa neuvoimme alkuun, tällä kertaa tarjolla on hieman edistyneempää rakentelua NodeMCU-alustan avulla.

Teksti: Jarno Lehtinen Kuvat: Sakari Leppä, Jarno Lehtinen

Srollin numerossa 2017.1 kävimme läpi NodeREDiä, jolla IoT-laitteita on kätevä yhdistää kokonaisuuksiksi. Jutussa tehtiin yksinkertainen IoT-laite ESP8266:sta Arduino IDE:llä. Tässä jutussa tutustumme tarkemmin ESP8266:een ja käymme läpi yhden vaihtoehdon esineverkon laitteiden lähtökohdaksi: NodeMCU ja Lua-ohjelmointikieli.

NodeMCU-kehitysalusta

NodeMCU on ESP8266:een perustuva IoT-kehitysalusta. Se pitää sisällään valmiin ESP8266-moduulin sisältävän piirilevyn ja siihen asennettavan laiteohjelmiston eli firmwaren. ESP8266 on halpa mikrokontrolleri, jossa on WLAN-ominaisuudet sisäänrakennettuna.

Tavallisiin ESP8266-moduuleihin verrattuna NodeMCU-kehitysalustassa on ESP8266:n lisäksi valmiina muun muassa USB-sarjaporttiadapteri ja jänniteregulaattori ESP:n tarvitsemaa 3,3 voltin käyttöjännitettä varten, sekä automaattinen ohjelmointitilaan siirtymisominaisuus sarjaportin RTS- ja DTR-signaalien avulla. ESP-moduulina siinä on ESP-12, jonka tärkein ero verrattuna yksinkertaisempiin ESP-01- ja ESP-201-moduuleihin on neljän

megatavun flash-muisti. ESP-01- ja ESP-201-moduuleissa on vain 512 kilotavun flash-muisti. NodeMCU:ta on mahdollista käyttää myös 512 kilotavun flash-muistilla, mutta pieni flash-muisti voi joissain tapauksissa aiheuttaa rajoitteita.

NodeMCU-kehitysalusta on yksinkertaisempia moduuleita huomattavasti helpompi ottaa käyttöön, koska NodeMCU:ssa on microUSB-portti, joten ohjelmointiin tarvitaan vain tietokone ja microUSB-kaapeli. NodeMCU:ssa on etuvastuksella varustettu LED valmiina nastassa D4, joten NodeMCU:n käytön voi aloittaa ilman muita lisähankintoja.

Kehitysalustan hankinta

NodeMCU-alustojen hinnat lähtevät verkkohuutokaupoissa postikuluineen noin kolmesta eurosta ylöspäin. Hyviä hakusanoja ovat NodeMCU, Lua ja ”development module board”. Jos haluaa fyysisesti erityisen pienen kehitysalustan, hyvä hakulauseke on ”WeMos D1 mini”. Joidenkin ESP-01-mallien kohdalla voi olla vähän harhaanjohtavasti maininta NodeMCU:sta, mutta selkeintä on varmistaa, että tuotteesta löytyy microUSB-portti. Tällöin se todennäköisesti sisältää kaikki tarvittavat toiminnot ja neljän megatavun

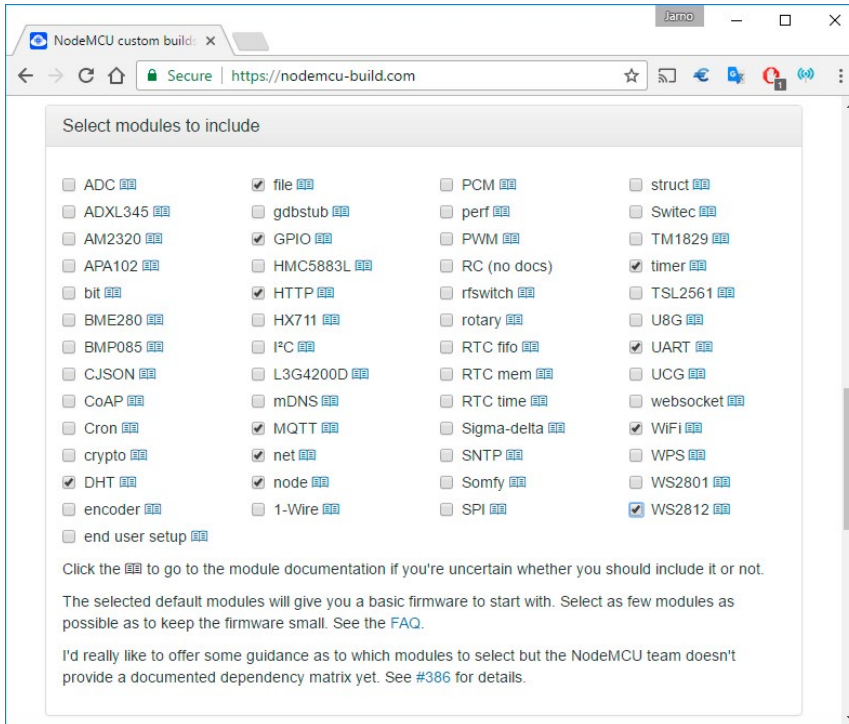
flash-muistin.

Firmwaren hankinta

NodeMCU-firmwarea ei ole valmiina ladattavana, vaan se räätälöidään tapauskohtaisesti. Hommaan on kuitenkin helppokäyttöinen www-sivu, jossa tarvitaan vain sähköpostiosoite valmista firmwarea varten. Kyseisellä sivulla tehdään valinnat, mitä ohjelmistomoduleita firmwareen halutaan mukaan. Kaikkia ohjelmistomoduleita ei kannata tai edes voi sisällyttää muistirajoitusten takia.

Mene siis sivulle nodemcu-build.com ja anna sähköpostiosoite. Perustoiminnot on valmiiksi ruksattu, mutta tässä jutussa käytetään perustoimintojen lisäksi *end user setup*-, *HTTP*-, *MQTT*- ja *WS2812*-moduuleita, joten ne kannattaa ruksata mukaan (ks. Kuva 1). Itse käytän usein *DHT*-lämpötila-/kosteusanturia, joten ruksin yleensä myös sen. Moduulien vieressä on pieni kirjasyntoli linkkinä moduulin tietoihin, ja monet moduuleista ovat sellaisia, että niiden tarpeen tietää selkeästi siinä vaiheessa kun on tekemässä jotakin. Jos esimerkiksi haluat keskustella NodeMCU:lla jonkin laitteen kanssa I²C:tä, ruksaa I²C jne.

”Start your build” -napin painamisen jälkeen menee muutamia minuut-



Kuva 1. Firmware-moduulit.

teja, jonka jälkeen sähköpostiin kolah-
taa kaksi firmware-tiedostoa. Niiden
koko on yleensä alle puoli megatavua
kappale. Toinen versioista on koko-
naislukuversio (integer) ja toisessa on
mukana liukuluvut (float). Liukuluku-
versio vie hiukan enemmän muistia,
mutta yleensä se ei ole ongelma, joten
tässä jutussa käytetään liukulukuja tu-
kevaa versiota.

NodeMCU-firmwaren asennus

Pitkään NodeMCU-firmwaren kir-
joittaminen muistiin oli melko moni-
vaiheinen juttu, mutta vuoden
2017 alussa tuli Windowsille erittäin
helppokäyttöinen ohjelma. Se löytyy
osoitteesta [github.com/marcelstoer/
nodemcu-pyflasher/releases](https://github.com/marcelstoer/nodemcu-pyflasher/releases). Ohjelma
ei vaadi mitään erityistä asennusta,
vaan sen voi käynnistää suoraan exe-
tiedostosta. (Ks. kuva 2.)

Nopeudeksi voi laittaa nopeimman
921 600 bps, ja koska NodeMCU-mo-
duuleissa on neljän megan flash-muisti,
flash-modeksi valitaan *dio*. Suositte-
len käyttämään flash-muistin tyhjennys-
optiota, koska monet ESP8266:n omat
sisäiset asetukset sijaitsevat flash-
muistilla ja ne nollautuvat vain tyh-
jentämällä koko flash-muistin. Vanhat
asetukset voivat joissain tapauksissa
aiheuttaa epävakautta ja kummallisia
ongelmia, jos vaikkapa jonkin van-
hemman tai täysin erilaisen firmwaren

asetukset ovat jääneet käyttöön.

Firmwaren flashäys onnistuu myös
esptool.py-nimisellä komentoriviltä
käytettävällä ohjelmalla, joka toimii
Windowsilla, Linuxilla ja MacOS:llä.
Se tarvitsee toimiakseen Pythonin. Jos
Python on asennettu, asentuu esptool
käskyllä

```
pip install esptool
```

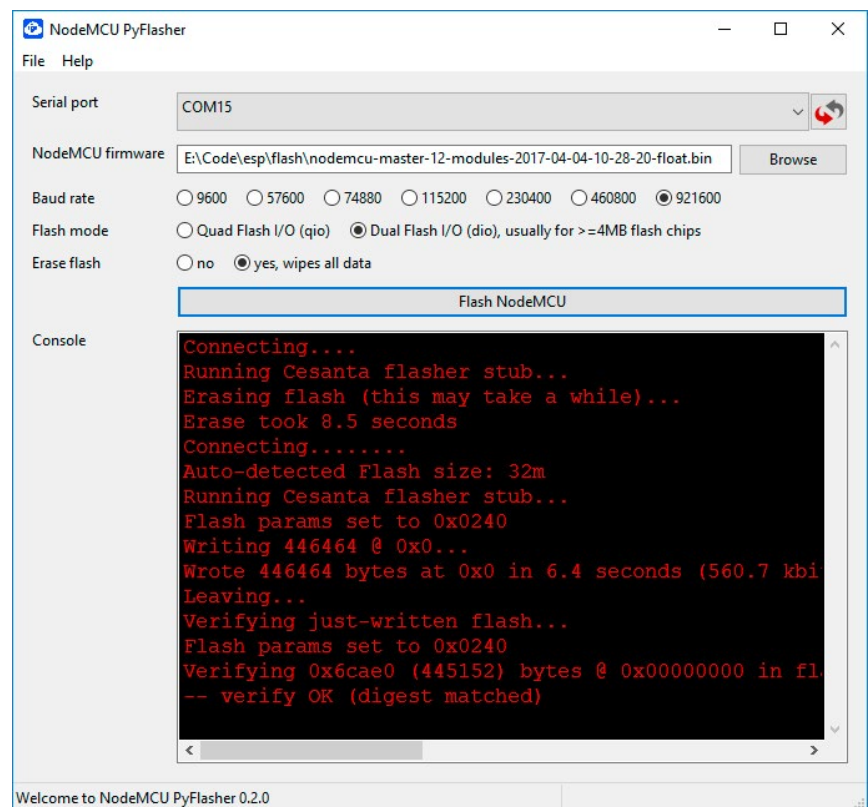
Flash-muistin tyhjennys ja firmwaren
kirjoitus tapahtuu seuraavilla komen-
noilla:

```
esptool.py --port /dev/ttyUSB0  
erase_flash  
esptool.py --port /dev/ttyUSB0  
write_flash 0x000000 firmware_  
tiedoston_nimi.bin
```

IFTTT-nappi ESPlorerilla

Vaikka NodeMCU:ta voisi periaattees-
sa käyttää pelkästään tavallisella termi-
naaliohjelmalla, on ohjelmointiin kä-
tevämpi käyttää ohjelmointityökalua.
ESPlorer on käytetyin ja monipuolisin
ilmainen NodeMCU-ohjelmointityö-
kalu. Ohjelman voi ladata osoitteesta
esp8266.ru/esplorer/. Alussa ESPlorer
saattaa vaikuttaa sekavalta ja omalaa-
tuiselta, mutta siitä tarvitaan normaali-
käytössä vain muutamia toimintoja
(ks. Kuva 3).

Valitse NodeMCU:n sarjaportti ja
paina sarjaporttivalinnan alapuolella
olevaa Open-nappia. Jos kirjoitit juuri
firmwaren, NodeMCU alustaa flash-
muistissa olevan tiedostojärjestelmää
noin kymmenen sekuntia, eikä vastaa
tänä aikana käskyihin. Kun yhteys on
syntynyt ja tiedostojärjestelmä alustet-
tu, voit antaa käskyjä suoraan alhaalla



Kuva 2. PyFlasher-ohjelma.

olevaan tekstikenttään. Voit kirjoittaa sinne esimerkiksi

```
print("Hei Skrolli!")
```

jonka NodeMCU tulostaa sarjaporttikonsoliin. Tämä vastaa samaa kuin että ottaisit yhteyden tavallisella terminaaliohjelmalla ja antaisit käskyjä sen kautta.

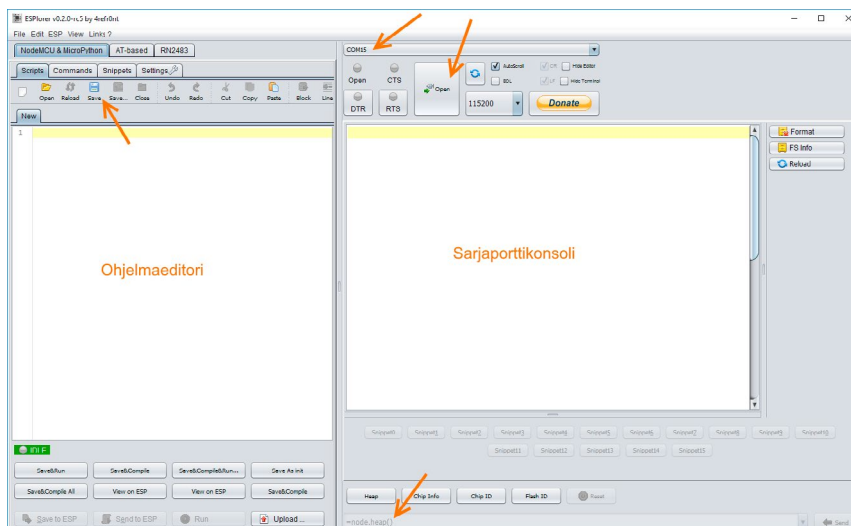
Tällä kertaa jätämme ledin vilkuttelun myöhemmäksi ja teemme suoraan hyödyllisen ohjelman. Ohjelma käyttää IFTTT-palvelua, jonka käyttöä käytiin läpi Skrollin numerossa 2017.1 NodeRED-jutun yhteydessä.

Edellisen numeron jutun kirjoittamisen jälkeen muutama IFTTT:n käyttämä termi on muuttunut. Reseptit ovat nykyään Appletteja ja Maker-palvelun nimi on Maker Webhooks. Jos et vielä ole käyttänyt IFTTT:tä, mene osoitteeseen ifttt.com ja rekisteröidy. Tee uusi IFTTT-Applet, jonka *This*-toiminnoksi valitse *Maker Webhooks* ja laukaisimeksi *Receive a web request*. Anna tapahtumalle nimi *nappi*. *That*-toiminnolle voit valita minkä tahansa haluamasi toiminnon. Itse käytän testaukseen yleensä *Google Driveä* ja *Add row to spreadsheet* -toimintoa.

Kirjoita tai kopioi seuraava ohjelma ohjelmaeditoriin:

```
local nappi = 3 -- NodeMCU:n
Flash-nappi
local event = "nappi" -- IFTTT:n
tapahtuman nimi
local key =
"xxxxxxxxxxxxxxxxxxxxxxx" --
henkilökohtainen IFTTT-avain
local lukko = false --
"debounce"-muuttuja
gpio.mode(nappi, gpio.INT) --
I/O-pinnin keskeytys päälle
gpio.trig(nappi, "down",
function() -- kun nappia
painetaan, suoritetaan HTTP-
pyyntö IFTTT:lle
    if not lukko then
        http.get("http://maker.
ifttt.com/trigger/" .. event ..
"/with/key/" .. key)
        print("Nappi!")
        lukko = true
        tmr.alarm(0, 1000, tmr.
ALARM_SINGLE, function()
            lukko = false --
vapautetaan lukko 1s päästä
        end)
    end
end)
```

Korvaa *key*-muuttuja henkilökohtaisella IFTTT-avaimellasi. Avaimen



Kuva 3. ESP8266-ohjelmisto.

saat esimerkiksi IFTTT:n sivuilta painamalla oikeasta ylänurkasta käyttäjän nimeä ja valitsemalla avautuvasta valikosta *Services*. Valitse sitten *Maker Webhooks* ja oikeasta ylänurkasta *Settings*. Avain on URL:ssä näkyvä merkijono.

Kun ohjelma on kirjoitettu, paina *Save*-nappia, jonka jälkeen ESP8266 kysyy tiedostolle nimeä. Anna tiedostolle nimi *init.lua*. Ohjelma tallentuu tietokoneelle, mutta lisäksi ESP8266 tallentaa sen myös ESP8266-moduulin flash-muistiin ja lopuksi ajaa ohjelman.

Voit kokeilla nyt painaa NodeMCU-moduulin flash-nappia, mutta tuloksena on muutaman sekunnin päästä sarjaporttikonsoliin ilmestyvä virheilmoitus *HTTP client: DNS failed for maker.ifttt.com*. Emme nimittäin ole vielä määrittäneet langattoman verkon asetuksia. Anna kaksi asetuskäskyä ESP8266-moduulin alareunassa olevaan tekstikenttään:

```
wifi.setmode(wifi.STATION)
wifi.sta.config("wlan_verkkosi_nimi", "wlan_verkkosi_salasana")
```

Ensimmäinen käsky määrittää ESP8266:n ottamaan yhteyttä tukiasemaan, eikä toimimaan itse tukiasemana. Asetukset tallentuvat ESP8266-moduulin flash-muistiin, joten asetukset tarvitsee antaa vain kerran tai kun Wi-Fi-verkon asetuksia halutaan muuttaa.

Nyt flash-napin painalluksen pitäisi laukaista IFTTT:hen määrittelemäsi toiminto, esimerkiksi lisätä rivi Google Docs -taulukkolaskentatiedostoon. Wemos D1 minissä ei ole valmiina flash-nappia, joten sen kanssa pitää

yhdistää hetkellisesti D3-nasta maahan. Maanasta on merkitty piirilevyllä G-kirjaimella.

ESP8266 ja langattoman verkon asetukset

Olen huomannut, että yksi langattoman IoT-laitteen käyttökelpoisuuteen eniten vaikuttavista tekijöistä on langattoman verkon nimen ja salasanan syöttö ja hallinta. Peruskokeilussa ne on helppo kirjoittaa osaksi varsinaista ohjelmaa, mutta ajatus siitä, että laitteen siirto tai vaikkapa langattoman verkon salasanan vaihto vaatisi koodimuutoksen ja uudelleenohjelmoinnin, tuntuu ikävältä. Siksi muutama sananen langattoman verkon asetuksista.

Monesti netistä löytyvissä esimerkeissä langattoman verkon asetukset on kovakoodattu ohjelmaan, eli kirjoitettu suoraan ohjelmalistaukseen. Tämä on yleensä turhaa, koska ESP8266 tallentaa asetukset flash-muistiin. Kolme ehkä yleisintä tapaa ohjelmoida ESP8266:tta ovat AI-Thinker (nk. AT-firmware), Arduino IDE ja NodeMCU. Ne kaikki noudattavat ESP8266:n flash-muistin muistirakennetta ja tallentavat asetukset samaan paikkaan. Jos siis olet millä tahansa edellä mainituista jo kerran asettanut langattoman verkon asetukset, ne säilyvät flash-muistissa, ellei niitä erikseen tyhjennä. Tämä on hyvä tiedostaa jo tietoturvan kannalta, koska asetukset on helppo lukea esimerkiksi `esptool.py`:llä, jos IoT-laitteeseen pääsee fyysisesti käsiksi. Langattoman verkon asetukset vievät 128 tavua ja ne sijaitsevat osoitteessa, joka on 0x8000 tavua flash-muistin lopusta. 4 Mt:n

flash-muistilla varustetun ESP8266:n verkkoasetukset voi lukea käskyllä

```
esptool.py --port COM15 read_
flash 0x3fe000 128 wifiasetukset.
bin
```

Jos flash-muistin koko on 512 kt, vaihda osoitteeksi 0x7e000. Tekemällä 128 tavua pitkän tiedoston blank.bin, joka sisältää pelkkää 0xff:ää, pelkkien verkkoasetusten tyhjennys tapahtuu 4 Mt:n flash-muistista käskyllä

```
esptool.py --port COM15 write_
flash 0x3fe000 blank.bin
```

Asetusten antamiseen/muuttamiseen on muutamia erilaisia järjestelmiä, mutta NodeMCU:n tapauksessa homma hoituu yksinkertaisimmillaan tietokoneella ja terminaaliohjelmalla. NodeMCU-firmwaren sarjaportin oletusnopeus on 115200 bps. Sen voi tehdä myös kännykällä tai tabletilla ja muutaman euron OTG-adapterilla. Itse olen käyttänyt Android-kännykällä Serial USB Terminal -ohjelmaa. Verkkoasetukset muutetaan ja talletetaan käskyllä:

```
wifi.sta.config("wlan_verkkosi_
nimi", "wlan_verkkosi_salasana")
```

Hommaa voi yksinkertaistaa sisällyttämällä rivin

```
w = wifi.sta.config
```

init.lua-tiedostoon, jolloin asetusten muuttamiseen riittää

```
w("wlan_verkkosi_nimi", "wlan_
verkkosi_salasana")
```

Asetukset päivittyvät heti ja ESP8266 rupeaa välittömästi muodostamaan yhteyttä, eikä uudelleenkäynnistystä välttämättä tarvita.

Yksi verkkoasetusten hallintaan kehitetty järjestelmä on nimeltään SmartConfig. SmartConfig on tuotu ESP8266:lle TI:n CC3000:sta. Valmis SmartConfig-kirjasto löytyy Arduino IDE:lle, mutta NodeMCU:lla sen käyttö vaatii itse käännettyä firmwarea. SmartConfig perustuu kännykkäohjelmaan, jolla asetukset syötetään. Ohjelma löytyy Google Play -kaupasta esimerkiksi nimellä ESP8266 SmartConfig.

Kun SmartConfigia kokeilee, tuntuu kuin tapahtuisi jotain yliluonnollista. Kännykkä ei nimittäin siirry toiseen langattomaan verkkoon ja silti joten-

kin mystisellä tavalla tietoa siirtyy ESP8266:een, jolla ei kyseisessä tilanteessa ole langattoman verkon tietoja. Temppu perustuu ESP8266:n kykyyn haistella verkkoliikennettä, mutta miten salatusta langattoman verkon liikenteestä voisi saada mitään dataa ilman että tietää salausavainta?

Selitys on melko nerokas - informaatio on koodattu pakettien pituuteen. Kännykkäsovellus lähettää UDP-paketteja broadcast-osoitteeseen. ESP8266 ei pysty näkemään salatusta datasta edes pakettityyppiä, puhumattakaan paketin sisällöstä, mutta pakettien koon ESP8266 pystyy näkemään. Vaikka salaus vaikuttaa myös pakettien kokoon, niin silti salatun paketin kokoon pystyy vaikuttamaan lähetyspäässä riittävästi, jotta informaatiota saadaan siirrettyä pakettikokoa vaihtelemalla. Järjestelmässä on kaksi heikkoutta: ESP8266 ei pysty kuuntelemaan kuin 2,4 gigahertsin langatonta verkkoa ja periaatteessa kaikki langattoman verkon alueella olevat laitteet voivat saada salausavaimen selville, jos sattuvat olemaan asetushetkellä kuulolla.

Toinen vaihtoehto on luoda ESP8266:sta tukiasema www-palvelimella, johon voi ottaa yhteyden esimerkiksi puhelimella ja antaa langattoman verkon tiedot. NodeMCU:ssa tätä varten on olemassa enduser setup-moduuli. Kun toiminto on käynnistetty, kännykällä otetaan yhteyttä ESP8266:n luomaan langattomaan tukiasemaan ja sitten kännykän selaimella mennään mille tahansa verkkosivuille. Kaikki verkkosivuositteet johtavat ESP8266:n asetussivulle, jossa

tiedot voidaan syöttää. Tämä ratkaisu on käytössä monissa kaupallisissa tuotteissa ja se on siinä mielessä yksinkertaisin, että kännykän lisäksi mitään erillisiä johtoja tai ohjelmia ei tarvita. Olen huomannut, että tämä tapa on hieman epävakaa ja vaatii toisinaan useita yrityksiä, mutta aina asetusten teko on loppujen lopuksi onnistunut. Yksi ongelmista on se, että kun kännykkä huomaa, että ESP8266:n luoman tukiaseman kautta ei pääse internetiin, se saattaa vaihtaa automaattisesti internetiin johtavaan tukiasemaan.

I/O-pinnit

I/O-pinnien numerointi saattaa aiheuttaa joskus sekaannuksia. Numeroita on käytössä ainakin kolmea erilaista: ESP8266-piirin jalkojen numerointi, GPIO-numerointi ja NodeMCU-moduulin piirilevylle painettu D-numerointi. Taulukkoon on kerätty numeroiden vastaavuudet ja joidenkin pinnien kaksoismerkitys. Lua-ohjelmakoodissa käytetään samoja numeroita kuin on painettu NodeMCU-moduuliin.

Turvallisia valintoja ovat siis vain D1, D2, D5, D6, D7. Muutakin pinnejä voi yleensä käyttää, kunhan on tietoinen niiden rajoituksista.

I/O-pinnien jännite on 3,3 voltia. Täydellistä varmuutta siitä, sietävätkö I/O-pinnit 5 voltin signaalia, en ole löytänyt.



NodeMCU-moduulin merkintä (Lua)	GPIO (Arduino IDE)	
D0	GPIO16	Ei tue keskeytyksiä. Ei ylösvetustusominaisuutta. Ei PWM:ää. Käytetään syväunesta heräämiseen kytkemällä tämä RST:hen. Joissain moduuleissa user-nappi.
D1	GPIO5	
D2	GPIO4	
D3	GPIO0	Flash-moodi. Flash-nappi NodeMCU-moduulissa. Pitää olla ylhäällä bootatessa normaalisti.
D4	GPIO2	Pitää olla ylhäällä bootatessa. Sininen LED NodeMCU:ssa.
D5	GPIO14	
D6	GPIO12	
D7	GPIO13	Sarjaportin vuonohjausta käytettäessä CTS.
D8	GPIO15	Boottilähteen valinta. Pitää olla alhaalla bootatessa normaalisti. Sarjaportin vuonohjausta käytettäessä RTS.
D9	GPIO3	Sarjaportin RX.
D10	GPIO1	Sarjaportin TX.

Taulukko 1. I/O-pinnit.


```

print("init.lua - Skrolli")

local tiedosto = "ohjelma.lua" -- Sen tiedoston nimi, joka suoritetaan initin jälkeen
local nappi = 3 -- NodeMCU:n Flash-nappi
local led = 4 -- NodeMCU:n sininen led

wifi.setmode(wifi.STATION) -- varmistetaan, että ESP ei ole turhaan tukiasema

-- vilkutaan lediä merkiksi siitä, että ohjelma on käynnissä
gpio.mode(led, gpio.OUTPUT)
local Q=0
tmr.alarm(0, 100, tmr.ALARM_AUTO, function()
    gpio.write(led, Q)
    Q=1-Q
end)

-- asetetaan napille keskeytys, jolla voi keskeyttää init.lua:n suoritus
gpio.mode(nappi, gpio.INT)
gpio.trig(nappi, "down", function()
    gpio.mode(nappi, gpio.INPUT) -- poistetaan keskeytys käytöstä
    tmr.stop(0) -- pysäytetään ledin vilkutusajastin
    tmr.stop(1) -- pysäytetään verkkoyhteyden odotusajastin
    tmr.stop(2) -- pysäytetään WiFi-asetustukiaseman ajastin
    gpio.write(led, gpio.HIGH) -- sammutetaan led
    print("init.lua keskeytetty!")
end)

-- tarkistetaan 2 s välein, onko verkkoyhteys saatu
tmr.alarm(1, 2000, tmr.ALARM_AUTO, function()
    if wifi.sta.getip() then
        tmr.stop(0) -- pysäytetään ledin vilkutusajastin
        tmr.stop(1) -- pysäytetään verkkoyhteyden odotusajastin
        tmr.stop(2) -- pysäytetään WiFi-asetustukiaseman ajastin
        gpio.write(led, gpio.HIGH) -- sammutetaan led
        print("IP: " .. wifi.sta.getip())
        print("Suoritetaan tiedosto \"" .. tiedosto .. "\"")
        dofile(tiedosto) -- jos verkkoyhteys on saatu, suoritetaan tiedosto
    end
end)

-- jos verkkoyhteyttä ei ole saatu 10 s aikana, käynnistetään WiFi-asetustukiasema
tmr.alarm(2, 10000, tmr.ALARM_SINGLE, function()
    print("Käynnistetään enduser setup")
    tmr.interval(0, 500) -- hidastetaan merkiksi ledin välkkymistä
    enduser_setup.start(function()
        tmr.alarm(3, 2000, tmr.ALARM_SINGLE, function()
            node.restart() -- 2 s päästä asetusten talletuksesta bootataan
        end)
    end)
end)
end)

```

Listaus 1. `init.lua`.

init.lua on uusi autoexec.bat

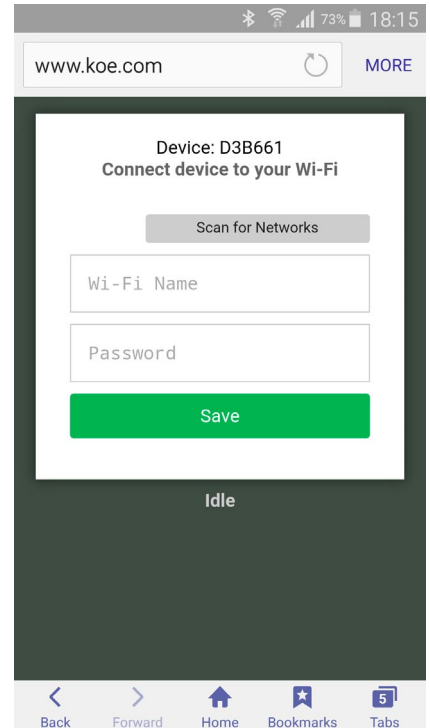
NodeMCU-firmware käynnistää autoomaattisesti `init.lua`-nimisen tiedoston. Jos `init.lua`:an sattuu ohjelmoimaan sellaisen virheen, että ESP8266 jumiuuu tai uudelleenkäynnistyy, voi olla vaikeaa saada sitä keskeytettyä. Siksi siihen on hyvä ohjelmoida jokin keskeytysmekanismi. Tässä on valmis `init.lua`, jonka ajo voidaan keskeyttää 2 sekunnin aikana painamalla flash-nappia tai yhdistämällä pinni D3 maihin. Tässä ohjelmassa varmistetaan myös se, että verkkoyhteys on kunnossa, ennen kuin varsinaista ohjelmaa kutsutaan.

Jos verkkoyhteyttä ei saada kymmenen sekunnin aikana, siirrytään tukiasematilaan, johon voi ottaa yhteyden esimerkiksi kännykällä. Valitse kännykän langattoman verkon asetuksista tukiasemaksi `SetupGadget_XXXXXX` ja mene selaimella mihin tahansa

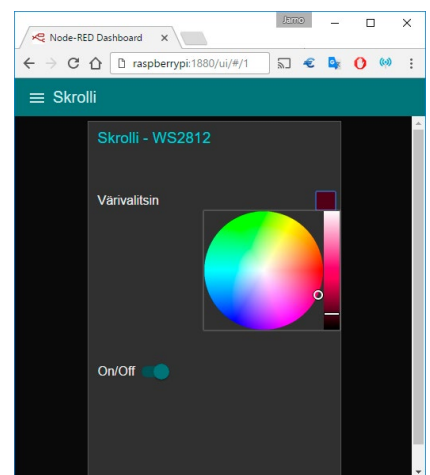
osoitteeseen, esimerkiksi `www.koe.com`. Nyt voit valita tai syöttää langattoman verkon nimen ja salasanan (ks. Kuva 4). Toisinaan toimintoa pitää yrittää muutamia kertoja, koska kännykkä saattaa esimerkiksi itsepin-taisesti vaihtaa tukiaseman joksikin toiseksi. Automaattisen yhteyden vaihdon saa yleensä kännykän asetuksista pois päältä.

NodeMCU, MQTT, LED-valonauha ja Node-RED

Lopuksi teemme ohjelman, jolla epäsuorassa valaistuksessa käytettävien sisustusvalojen väriä ja kirkkautta voi säätää verkkosivun kautta (ks. Kuva 5). Noin 10 eurolla saa verkosta 50 RGB-ledin nauhoja, joiden ohjaus perustuu WS2811- tai WS2812-piiriin. Ohjelmoinnin kannalta WS2811:llä ja WS2812:lla ei ole eroa. Hakusanoina toimii 50pcs, ws2811 ja string. Myös



Kuva 4. Wi-Fi-asetusten teko kännykällä.



Kuva 5. LED-nauhan ohjain.

monet Adafruitin NeoPixel-tuotteet perustuvat jompaankumpaan näistä piireistä. LED-nauhoja voi ketjuttaa, ja ohjelmakoodissa on ledien määräksi määritelty 100 kpl. Huomaa, että 50 lediä voi viedä täydellä kirkkaudella reilusti yli yhden ampeerin virran, joten huolehdi, että virtalähde on riittävä. LED-nauha vie vain yhden I/O-pinnin, ja NodeMCU-firmwareassa se on kiinteästi D4. Samaan aikaan ei siis voi ohjata NodeMCU-moduulissa olevaa D4-pinniin kytkettyä sinistä lediä.

Tallenna ohjelma nimellä `ohjelma.lua`. Skrollin numerossa 2017.1 käytiin läpi Node-RED ja MQTT. Koodi on suoraan yhteensopiva Node-RED:n Dashboard-switch- ja Dashboard-co-

```

local mqtt_broker = "xxxxx.xxxxx.xxx"
local mqtt_port = 1883
local mqtt_user = "xxxxx"
local mqtt_pwd = "xxxxx"
local mqtt_clientid = "NodeMCU-" .. string.format('%x', node.chipid())
local mqtt_topic = "NodeMCU"
local nappi = 3

-- varmistetaan, ettei oteta yhteyttä useaan kertaan päällekkäin
if not m then
  m = mqtt.Client(mqtt_clientid, 120, mqtt_user, mqtt_pwd)
else
  m:close()
end

-- muodostetaan yhteys mqtt-brokeriin
m:connect(mqtt_broker, mqtt_port, 0, 1, function(conn)
  print("Yhteys mqtt-brokeriin muodostettu!")
  m:publish("status", mqtt_clientid .. " yhdistetty!", 0, 0)
  m:subscribe(mqtt_topic, 0, function(conn)
    print("Tilattu " .. mqtt_topic .. "-topic")
  end)
  m:on('message', function(conn, topic, viesti)
    print("mqtt-viesti - topic:" .. topic .. ", viesti:" .. viesti)
    if viesti:upper() == "PING" then
      m:publish("status", mqtt_clientid .. " vastaa pingiin!", 0, 0)
    else
      viestinkäsittelija(topic, viesti)
    end
  end)
end)

-- asetetaan napille keskeytys ja sen käsittelijä
local lukko = false
gpio.mode(nappi, gpio.INT)
gpio.trig(nappi, "down", function()
  if not lukko then
    m:publish("status", mqtt_clientid .. " - Nappi!", 0, 0)
    print("Nappi!")
    lukko = true
    tmr.alarm(0, 1000, tmr.ALARM_SINGLE, function()
      lukko = false
    end)
  end
end)

local r = 100
local g = 100
local b = 100
-- täällä käsitellään käyttäjän mqtt-viestit
function viestinkäsittelija(topic, viesti)
  if viesti:sub(1, 1) == "#" then
    r = tonumber(viesti:sub(2, 3), 16)
    g = tonumber(viesti:sub(4, 5), 16)
    b = tonumber(viesti:sub(6, 7), 16)
    buffer:fill(r, g, b)
  elseif viesti:upper() == "TRUE" then
    buffer:fill(r, g, b)
  elseif viesti:upper() == "FALSE" then
    buffer:fill(0, 0, 0)
  end
  ws2812.write(buffer)
end

-- tehdään alustukset lednauhaa varten
if not buffer then
  ws2812.init()
  buffer = ws2812.newBuffer(100, 3)
end

```

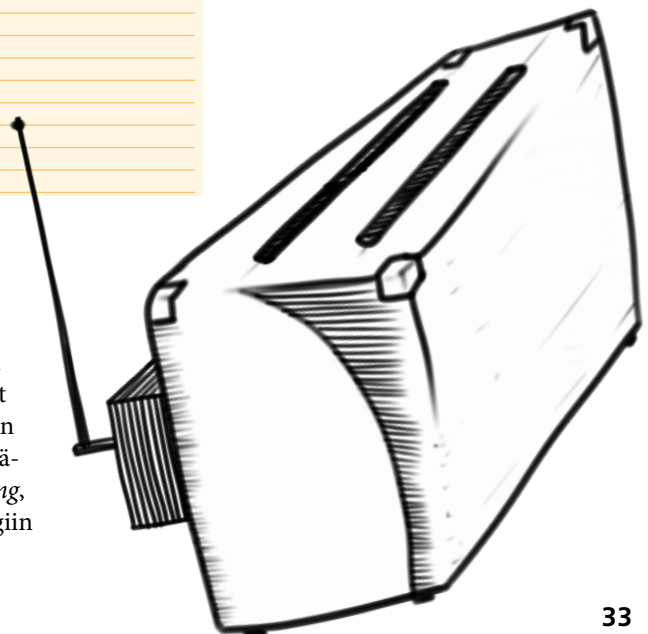
Listaus 2. ohjelma.lua.

lour picker -nodejen kanssa. Jos määrittelit MQTT-nodeen *Retain: true*, viimeisin väri viesti tulee aina NodeMCU-moduulille sen bootatessa, joten sitä voi käyttää eräänlaisena tallennuksena. Colour picker -moduuli antaa oletuksena payloadiksi värin RGB-heksadesimaalimuodossa #1f2e3d. Lisäksi valot voidaan sammuttaa tai sy-

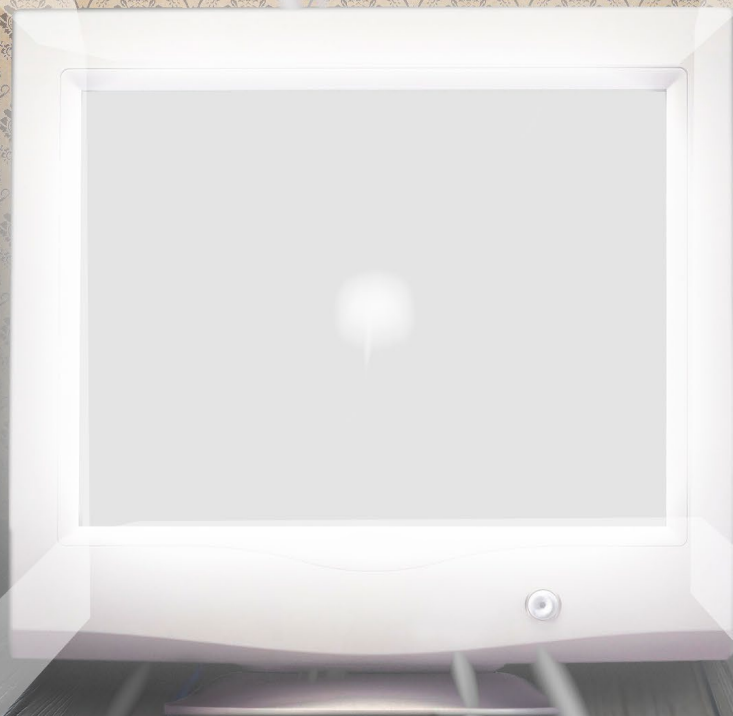
tyttää lähettämällä true tai false, jotka ovat switch-noden oletusarvot. Käskyt lähetetään topicin *NodeMCU*. Kun lähetetään käsky *ping*, laite vastaa pingiin *status*-topicissa.

Lopuksi teemme ohjelman, jolla epäsuorassa valaistuksessa käytettävien sisustusvalojen väriä ja kirkkautta voi säätää verkkosivun kautta. Noin 10 eurolla saa verkosta 50 RGB-ledin nauhoja, joiden ohjaus perustuu WS2811- tai WS2812-piiriin. Ohjelmoinnin kannalta WS2811:llä ja WS2812:lla ei ole eroa. Hakusanoina toimii 50pcs, ws2811 ja string. Myös monet Adafruitin NeoPixel-tuotteet perustuvat jompaankumpaan näistä piireistä. LED-nauhoja voi ketjuttaa, ja ohjelmakoodissa on ledien määräksi määritelty 100 kpl. Huomaa, että 50 lediä voi viedä täydellä kirkkaudella reilusti yli yhden ampeerin virran, joten huolehdi, että virtalähde on riittävä. LED-nauha vie vain yhden I/O-pinnin, ja NodeMCU-firmwaressa se on kiinteästi D4. Samaan aikaan ei siis voi ohjata NodeMCU-moduulissa olevaa D4-pinniin kytkettyä sinistä lediä.

Tallenna ohjelma nimellä *ohjelma.lua*. Skrollin numerossa 2017.1 käytiin läpi Node-RED ja MQTT. Koodi on suoraan yhteensopiva Node-RED:n Dashboard-switch- ja Dashboard-colour picker -nodejen kanssa. Jos määrittelit MQTT-nodeen *Retain: true*, viimeisin väri viesti tulee aina NodeMCU-moduulille sen bootatessa, joten sitä voi käyttää eräänlaisena tallennuksena. Colour picker -moduuli antaa oletuksena payloadiksi värin RGB-heksadesimaalimuodossa #1f2e3d. Lisäksi valot voidaan sammuttaa tai sytyttää lähettämällä true tai false, jotka ovat switch-noden oletusarvot. Käskyt lähetetään topicin *NodeMCU*. Kun lähetetään käsky *ping*, laite vastaa pingiin *status*-topicissa. 🐛



Muistatko, kun monitori säteili?



Ruotsista alkunsa saanut säteilypaniikki toi TCO-merkinnät putkinäyttöihin.

Teksti: Vesa Linja-aho Kuvat: Manu Pärssinen, Jari Jaanto

Kun lapsuudenkotiini ostettiin vuonna 1992 ensimmäinen tietokone, se oli jo hieman vanhahtava 286-kone, ja hintaa oli muistikuvani mukaan 5000 markkaa. Neljäsluokkalaisen rillipään huomio kiinnittyi koneen VGA-kuvaputkinäytössä komeilevaan tekstiin *Low emission*. Äiti tiesi kertoa, että se tarkoittaa matalaa säteilytasoa.

Pari vuotta myöhemmin kone vaihtui 486-malliin, jonka säteilymerkinnöistä ei jäänyt mitään muistikuvaa. Sen sijaan vuonna 1997 Celeron-myllyssä olikin jo TCO-merkki.

Low emission -mainoslauseen ja TCO-merkin taustalla oli 1980-luvun lopulla Ruotsista alkunsa saanut säteilypaniikki. Luulin pitkään TCO:ta joksikin viranomaiseksi tai testauslaitokseksi, mutta myöhemmin selvisi,

että lyhenne tulee sanoista *Tjänstemännens centralorganisation*, virkamiesten keskusliitto. Kyseessä oli ammattijärjestön lanseeraama tarra, joka kertoi vähäisestä säteilystä.

Jo pikkulapsena 1980-luvun lopulla äiti varoitteli katsomasta televisiota liian läheltä, ”ettei mene silmät pilalle” ja ”siitä tulee säteilyä”. Talvi- ja jatkosodista hengissä selvinnyttä isää eivät taas kodinelektronikan säteilemiset voineet vähempää kiinnostaa.

Televisioissa ja tietokoneiden näyttöpäätteissä käytettiin vielä viime vuosikymmenelle asti katodisädeputkea, joka perustui elektronisuihkun loisteaineelle piirtämään kuvaan. Elektroneja kiihdytettiin jännitteellä, joka isoissa näytöissä oli korkeimmillaan parikymmentä kilovolttia. Näin suurella kiihdytysjännitteellä todella

syntyi aavistuksen verran röntgensäteilyä, josta siitäkin lähes kaikki imeytyi kuvaputken lasiin. Määrä on mitätön verrattuna röntgentutkimuksista ja lentomatkoilta saatuun säteilyyn, jatkuvasta luonnon taustasäteilystä puhumattakaan.

Mistä puheet television ja tietokonenäyttöjen säteilyvaarasta sitten saivat alkunsa?

Televisio säteili sanomalehdessä jo 1956

Vanhin maininta television säteilystä löytyy Turun Sanomista 14.6.1956, jossa kansalaisia rauhoitellaan ydinkokeiden säteilyvaarasta: pikku-uutisen mukaan televisiolaitteista säteilee vähäinen määrä röntgensäteitä, mutta ne eivät ole vaarallisia. Uutta kodinkonetta käytettiin uutisessa pikemminkin

säteilypelkojen lievittämiseen kuin sensaation hakemiseen: ydinkokeista tulee huomattavasti pienempi säteilyannos kuin vaarattomiksi miellettyä ja intoa ja innostusta herättäneistä televisiovastaanottimista, jotka vielä 1950-luvulla olivat harvojen herkkua. Muina vertailukohteina käytettiin itsevalaisivia kellotauluja ja jalan mittaus- ta röntgenillä kenkiä varten.

Reilun kuukauden päästä 18.7.1956 sama lehti tosin varoittaa hieman sensaatiohakuisesti asiantuntijan suulla, että *korkeajännitteiset televisiovastaanottimet saattavat olla vaarallisia. Vaikutukset tulevat esille seuraavissa sukupolvissa, joiden lapsilla saattaa olla epäterveitä luonteenominaisuuksia.* Tavallisista televisioista ei lehden mukaan tarvitse olla niinkään pelossa, sillä niiden säteily määrä on pieni, mutta *voimakkaat vastaanottimet ovat eri asia.*

Mikään suosikkiaihe television säteily ei lehdistölle ollut: seuraava maininta Kansalliskirjaston digitoiduista sanomalehtiarkistoista löytyy vasta kuuden vuoden päästä. Länsi-Savo 24.5.1962 kirjoittaa, että atomienergiakomission jäsen, tohtori Robert Wilson kertoo, että USA:n ja Neuvostoliiton ydinkokeiden aiheuttama säteily on vähemmän kuin television aiheuttama säteily. Eli vanhan toistoa, mutta eri asiantuntijan suulla.

Tarkoituksena oli siis rauhoitella kylmän sodan ydinkokeiden hermostuttamia kansalaisia, mutta samalla jäi elämään käsitys siitä, että televisio säteilee. Sosiaalista mediaa ei tuolloin ollut, vaan tieto kulki suusta suuhun: joku muisti lukeneensa lehdestä television säteilyvaarasta, eikä ollut nettiarkistoa, josta tarkistaa asian laita, joten rikkinäinen puhelin -ilmiö todennäköisesti muutti ja paisutti tarinaa matkalla.

Näyttöpäätetyö yleistyy

Television säteilyvaara ei sen koommin nostanut mediassa päätään, mutta 1970-luvun loppupuolella esiin marsi uusi mörkö: tietokoneet korvasivat paperitöitä suurissa yrityksissä ja valtionhallinnossa. Televisioiden säteilyvaarasta suusta suuhun levinneet jutut yleistettiin – sinänsä ihan loogisesti – näyttöpäätetyöhön. Olihan käytetty tekniikka, katodisädeputki, sama.

Varhaisin maininta näyttöpäätetyön säteilyhuolista löytyy Etelä-Suomen

Sanomista 28.10.1980. Lehti raportoi työterveyspäiviltä näyttöpäätetyön todellisista ja kuvitteellisista haitoista: Työterveyslaitoksen asiantuntija laskee kuviteltuihin haittoihin näyttöpäätteen säteilyn. Nykyiset ja aiemmat tutkimustulokset osoittavat asiantuntijan mukaan vakuuttavasti, etteivät päätteet lähetä haitallista säteilyä. Työsuojelumielessä enemmän huolestuttivat tuki- ja liikuntaelinvaivat sekä – nykykäyttäjillekin niin tutut – laitteiston käyttöhäiriöiden, katkosten ja odotusten aiheuttamat psykologiset haitat.

Pian tämän jälkeen Länsi-Savo 22.3.1981 kertoo artikkelissaan ”Konekumous konttoreissa”, kuinka *sulkakynän on korvannut näyttöpäätteen ja mikrokortti, ja tietokonemuisti on nielaisut mapit.* Näyttöpäätteen määrä lähenteli jo 50000:ta. Huolestuneita oltiin lähinnä työpaikkojen vähenemisestä: artikkelissa mainitaan lyhyesti työsuojeluongelmista, että huhut [näyttöpäätteen] vaarallisesta säteilystä on jo kumottu.

Ruotsista kajahtaa

Muutaman vuoden oli säteilyrintamalla hiljaista, mutta näyttöpäätetyö yleistyi 1980-luvulla rajusti, mikä oli omiaan nostamaan ilmiön taas tapetille. Länsi-Savo 6.2.1986 uutisoi, että Tanskassa tutkitaan näyttöpäätteen vaarat. Tanskan liike- ja toimistotyöntekijöiden liitto ryhtyi tutkimuksessaan selvittämään näyttöpäätteen yhteyttä keskenmenoihin ja raskaushäiriöihin. Taustalla olivat tutkimukset, joissa ”päätteen magneettisäteily” oli todettu aiheuttavan rotille ja hiirille keskenmenoja ja sikiövaurioita.

Tutkimuksen tuloksista ei uutista kuulunut. Tämän jälkeen katodisädeputken säteily mainitaan Länsi-Savon jutussa 1.5.1989, jossa mainitaan myös Ruotsissa perustettu sähköallergikko-

jen yhdistys. Näyttöpäätteen sähkö- ja magneettikenttien raskausriskeistä todetaan, että valtaosassa tutkimuksia väitetään todetun kiistatta, ettei päätetyö haittaa sikiön kehitystä tai aiheuta keskenmenoja. Ruotsissa ja Yhdysvalloissa taas on julkaistu raportteja, joissa ”keskenmenon mahdollisuuden on todettu lisääntyneen”.

Samassa lehdessä 29.7.1991 lääkäri vastaa huolestuneelle äidille television vaaroista Länsi-Savon ”Seuraava potilas” -palstalla, että jos jotain terveyshaittoja on, ne johtuvat ohjelmista tai kyhjöttämisestä, ja säteilyn puolesta televisio on turvallinen.

Nykykonsensus: ei vaaroja

Kuvaputkinäyttöjen haittoja ei enää kannata tutkia, kun teknologia on poistunut käytöstä. Mitään haittoja ei vuosikymmentenkään saatossa löydetty. ”Jo vuonna 1968 USA:ssa säädettiin yläraja katodisädeputkien säteilylle. Käytännössä putket säteilivät turvarajoja vähemmän. Säteilyannos jäi pienemmäksi kuin luonnon taustasäteily. Lisäksi laajat tutkimukset eivät ole löytäneet mitään haitallisia terveysvaikutuksia katodiputkista”, kertoo säteilyn terveysvaikutuksia tutkinut fysiikan professori **Tapio Ala-Nissilä** Skrollille.

Ammattiliiton TCO-tarrat ja *low emission* -tekstit todennäköisesti olivatkin toimiva lääke näyttöpäätteen ilmiöön: kun käyttäjä uskoi, että näyttö täyttää tiukat säteilynormit, oireetkin häipyvät. Vastaavaa temppua on kehitetty eräässä suomalaiskorkeakoulussakin, tosin modernimman väivanaiheuttajan eli wlan-tukiaseman kanssa: tukiaseman aiheuttamat oireet erääseen opiskelijaan katosivat, kun tukiasema ”sammutettiin” käyttämällä sivuleikkureita sen merkkivaloleidin jalkoihin. 🗑️



STRINGOLOGIA

– bioinformatiikan kivijalka

Merkkijono kuuluu lähes kaikkien ohjelmointikielien perustietorakenteisiin. Niiden tehokas käsittely on kuitenkin kaukana yksinkertaisesta, ja siitä onkin kasvanut kokonainen tutkimusala nimeltä stringologia.

Teksti: Jarno Niklas Alanko Kuvat: Mikko Torvinen, Jarno Niklas Alanko, Nasu Viljanmaa

Yleinen ongelma tietojenkäsittelyssä on selvittää, löytyykö merkkijono S jostain kohtaa merkkijonoa T , ja jos löytyy, niin mistä kohdista. Tämä ongelma tulee vastaan esimerkiksi tekstinkäsittelyohjelmassa, kun käyttäjä painaa Ctrl-F-näppäinyhdistelmää. Ei kuulosta kovin monimutkaiselta ongelmalta, eihän? Valitettavasti tehtävä ei ole niin helppo kuin ensivaikutelma antaa ymmärtää.

Yksinkertaisin tapa ratkaista asia on kirjoittaa kaksi sisäkkäistä for-silmukkaa, jotka kokeilevat kaikki mahdolliset aloituskohdat läpi (ks. listaus 1). Ongelma on, että pahimmassa tapauksessa tähän kuluu aikaa suhteessa lukuun nm , jossa n ja m ovat merkkijonojen T ja S pituudet. Algoritmin keskimääräinen aikavaativuus on siis merkittävästi $O(nm)$. Jos n ja m ovat molemmat suuria, niin tämä jumittaa tietokoneen nopeasti. Stringologian keskeisiä ongelmia on tehokkaampien algoritmien kehittäminen tähän ongelmaan ja kaikkiin sen variaatioihin

Ensimmäisen $O(nm)$ -algoritmia pa-

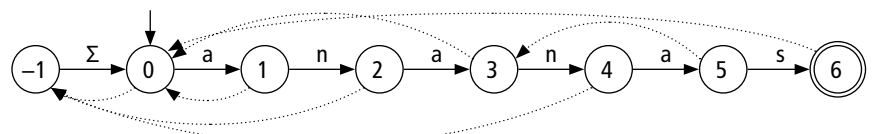
remman algoritmin julkaisivat tieltävästi **Donald Knuth**, **Vaughan Pratt** ja **James Morris** vuonna 1977. Heidän menetelmänsä tunnetaan nykyään KMP-algoritmina julkaisijoiden sukunimien alkukirjaimien mukaan, ja se on perusta monille edistyneemmille algoritmeille. Algoritmi rakentaa merkkijonosta S äärellisen tilakoneen (finite state machine), joka koostuu tiloista ja siirtymistä (ks. kuva 1). Rakennetulla automaatilla voi hakea merkkijonoa S mistä tahansa merkkijonosta T . Automaatin rakentaminen vie tehokkaalla algoritmilla $O(m)$ aikaa, ja merkkijonon T läpikäynti $O(n)$ aikaa, joten algoritmi vie yhteensä $O(n+m)$ aikaa, mikä on huima paran-

nus naviin $O(nm)$ -algoritmiin.

Sittemmin ongelmaan on keksitty valtava joukko erilaisia $O(n+m)$ -aika-vaativuuden ratkaisuja. Nopeimmat välttävät jopa lukemasta kaikkia merkkijonon T merkkejä, jos ympäröivien merkkien perusteella on selvää, että merkkijono S ei voi löytyä tietyistä kohdista.

Loppuosataulukko kertoo merkkijonon syvimät salat

Sovelluksissa on yleistä, että käyttäjä tekee useita hakuja samaan tekstiin T . Näissä tapauksissa on hyödyllistä rakentaa indeksi merkkijonolle T , jonka avulla voidaan vastata kaikkiin kyselyihin S nopeasti.



Kuva 1. Knuth-Morris-Pratt automaatti merkkijonolle ananas. Alkutila on merkitty nuolella. Automaatti lukee merkkijonoa T merkki kerrallaan siten, että luettu merkki määrää automaatin seuraavan tilan. Jos luettu merkki vastaa seuraavan kaaren merkkiä, siirtyy algoritmi seuraavaan tilaan, muulloin algoritmi seuraa katkoviivalla merkittyjä nuolia, kunnes oikealle menevässä kaarissa on oikea merkki. Isolla sigmalla merkitty kaari sopii kaikkiin merkkeihin. Jos algoritmi päättyy tilaan 6, merkkijono S on löytynyt.

Listataan kaikki merkkijonot, jotka alkavat jostain kohtaa merkkijonoa T ja jatkuvat merkkijonon T loppuun saakka. Näitä kutsutaan merkkijonon T loppuosiksi eli suffiksiksi. Järjettään loppuosat aakkosjärjestykseen. Listauksessa 2 on esimerkkinä merkkijonon "alabalrabalrab" loppuosat aakkosjärjestyksessä. Nyt voidaan määrittellä merkkijonon T loppuosat eli suffiksitaulukko SA (Suffix Array): taulukon elementti SA[i] kertoo, mistä kohtaa merkkijonoa T alkaa aakkosjärjestyksessä i:s loppuosat.

Ensinäkemältä voi olla vaikea keksiä, mitä iloa tällaisesta taulukosta voi olla, mutta osoittautuu, että loppuosataulukko on avain valtavaan kirjoon tehokkaita merkkijonoalgoritmeja. Taulukon hyöty perustuu seuraavaan havaintoon. Haetaan vaikka merkkijonoa "bal" merkkijonosta "alabalrabalrab". Nyt on hyvä ymmärtää, että kaikki osamerkkijonot löytyvät jonkin loppuosan alusta. Aakkosjärjestetystä loppuosalistasta näkee, että ne loppuosat, jotka alkavat merkkijonolla "bal" ovat peräkkäin aakkosjärjestyksessä, tässä tapauksessa loppuosat numero 9, 10 ja 11. Tätä aluetta loppuosalistassa kutsutaan merkkijonon "bal" *leksikografiseksi intervalliksi*, ja sen voi paikallistaa binäärihakemalla ensin sen alkukohdan ja sitten loppukohdan. Merkkijonon "bal" esiintymien alkukohdat voidaan nyt lukea loppuosataulukon kohdista 9, 10 ja 11, joissa on arvot 9, 12 ja 4. Merkkijonon "bal" esiintymät tosiaan alkavat indekseistä 9, 12 ja 4.

Loppuosataulukon avulla voi myös selvittää esimerkiksi pisimmän useamman kerran esiintyvän osamerkkijonon merkkijonossa T. Esimerkissämme pisin toistuva merkkijono on "balrab". Pisimmän toistuvan osamerkkijonon etsimisen ongelma on sama kuin ongelma löytää pari loppuosaa, joilla on pisin mahdollinen yhteinen alkuosa. Kaikkien osamerkkijonojen joukko kun on kaikkien loppuosien al-

kuosien joukko. Lisäksi riittää vertailla vain aakkosjärjestyksessä peräkkäisiä loppuosia, koska mitä lähempänä kaksi loppuosaa ovat toisiaan aakkosjärjestyksessä listassa, sitä pidempi yhteinen alkuosa niillä on. Saadaan siis seuraava algoritmi: käydään läpi kaikki aakkosjärjestyksessä peräkkäiset loppuosaparit käyttäen loppuosataulukkoa ja raportoidaan pisin löytynyt yhteinen alkuosa.

Loppuosataulukon rakentaminen

Yksinkertaisin tapa rakentaa loppuosataulukko on muodostaa kaikki mahdolliset loppuosat, ja järjestää ne millä tahansa yleiskäyttöisellä järjestysohjelmalla kuten merge sortilla. Tähän tarvitaan $O(n \log n)$ loppuosien vertailua. Ongelma tässä on, että pahimmassa tapauksessa vertailut ovat työläisiä. Esimerkiksi jos syöte on n kappaletta pelkkää a-kirjainta, niin lopulliseksi aikavaativuudeksi tulee $O(n^2 \log n)$.

Hieman tehokkaampi tapa ratkaista ongelma on käyttää radix sort -tyyppistä järjestämistä merge sortin sijaan. Tässä loppuosat ryhmitellään ensiksi ensimmäisen merkin mukaan, syntyneet ryhmät toisen merkin mukaan, ja niin edelleen rekursiivisesti, kunnes kussakin ryhmässä on vain yksi loppuosat. Aikavaativuudessa päästään näin eroon logaritmisesta kertoimesta ja saadaan $O(n^2)$ -algoritmi. Loppuosien yhteispituus on $O(n^2)$, joten vaikuttaa siltä, että nopeampaa algoritmia ei voi olla. Yllättävää kuitenkin, tästä voi parantaa vielä paljon.

Tehokkaampaa algoritmia varten täytyy käyttää hyväksi sitä huomiota, että järjestettävät merkkijonot eivät ole mielivaltaisia, vaan kaikki ovat syöte-merkkijonon loppuosia.

Udi Manber ja **Gene Myers** julkaisivat vuonna 1990 ovelan algoritmin, jota voidaan pitää läpimurtona merkkijonotutkimuksessa. Se rakentaa ajassa $O(n \log^2 n)$ käänteisen loppuosatau-

lukon, jonka i:s alkio kertoo kohdasta i alkavan loppuosan järjestysnumeron aakkosjärjestyksessä. Käänteisen loppuosataulukon voi muuttaa helposti tavalliseksi loppuosataulukoksi – yksityiskohdat jääkööt lukijalle pohdittavaksi.

Sukellus syvemmälle algoritmiikkaan

Manberin ja Myersin algoritmi alustetaan siten, että korvataan merkkijonon S merkit kokonaisluvuilla, jotka säilyttävät aakkoston järjestyksen. Esimerkiksi merkkijonon "ananabanana" aakkosto on a, b ja n, joten voidaan korvata a:t yksöisillä, b:t kaksoisilla ja n:t kolmosilla. Näin saatu lista R kokonaislukuja on karkea arvio loppuosien keskinäisestä järjestyksestä: luvut kertovat loppuosien järjestyksen, jos jokaisesta otetaan huomioon vain ensimmäinen merkki.

Algoritmi perustuu 70-luvulla keksittyyn prefix doubling -tekniikkaan. Ensimmäisessä iteraatiossa tätä approksimaatiota tullaan parantamaan siten, että se ottaa huomioon kaksi ensimmäistä merkkiä jokaisesta loppuosasta, seuraavassa iteraatiossa neljä ensimmäistä, seuraavassa kahdeksan, ja niin edelleen, kunnes kaikki merkit on otettu huomioon.

Merkkijonon alabalrabalrab loppuosataulukko.

[16, 8, 3, 1, 10, 13, 5, 17, 9, 12, 4, 2, 11, 14, 6, 15, 7]

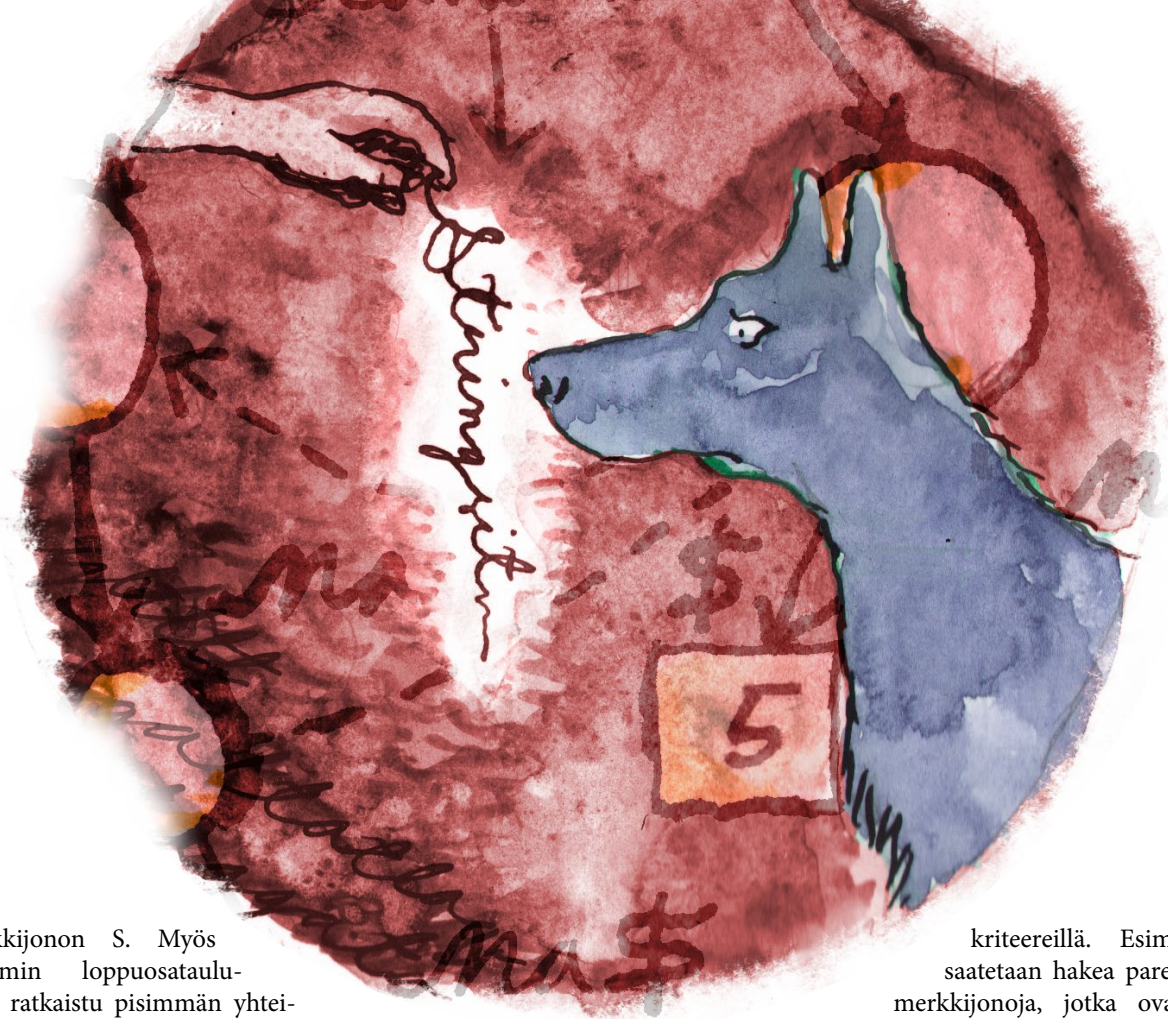
Merkkijonon alabalrabalrab loppuosat aakkosjärjestyksessä.

1	ab
2	abalbalrab
3	abalrabalbalrab
4	alabalrabalbalrab
5	albalrab
6	alrab
7	alrabalbalrab
8	b
9	balbalrab
10	balrab
11	balrabalbalrab
12	labalrabalbalrab
13	lbalrab
14	lrab
15	lrabalbalrab
16	rab
17	rabalbalrab

Listaus 2. Alabalrabalrab-loppuosataulukko.

```
vector<int> matches;
for(int i = 0; i < T.size()-S.size()+1; i++){
    bool good = true;
    for(int j = 0; j < S.size(); j++){
        if(S[j] != T[i+j]) good = false;
    }
    if(good) matches.push_back(i);
}
```

Listaus 1. Haetaan merkkijono S esiintymät merkkijonosta T.



merkkijonon S. Myös aiemmin loppuosataulukolla ratkaistu pisimmän yhteisen merkkijonon etsiminen onnistuu helposti – riittää vain käydä puu läpi ja palauttaa se sisäsolmu, joka on kaaimpana juuresta.

Suffiksipuu on myös ennen kaikkea hyödyllinen käsitteellinen työkalu merkkijono-ongelmien kanssa järjelliseen. Useimmille ongelmille löytyy luonnollinen tulkinta suffiksipuun näkökulmasta. Esimerkiksi jos halutaan laskea merkkijonon kaikki erilaiset osajonot, vastaus on kaikkien suffiksipuussa juuresta lähtevien polkujen määrä, koska joka jokainen osajono vastaa jotain juuresta alkavaa polkua ja kaikki juuresta alkavat polut vastaavat jotain osajonoa. Tässä tapauksessa polulla voidaan tarkoittaa myös sellaista polkua, joka pysähtyy johonkin keskelle kaaren osamerkkijonoa.

Uudet tuulet

Merkkijonoalgoritmien tutkimus menee eteenpäin kovaa vauhtia, ja Helsingin yliopisto on ilahduttavasti tutkimuksen eturintamassa. Jo vaikiintuneiden tietorakenteiden kuten loppuosataulukoiden ja -puiden ra-

kentamiseen kehitetään edelleen uusia algoritmeja, jotka on räätälöity toimimaan vähällä muistilla, rinnakkain tai mahdollisimman välimuistiystävällisesti.

Uusimpia tutkimussuuntia alalla ovat pakatut merkkijoindeksit, jotka pystyvät tekemään kaiken, mitä voisi tehdä suffiksipuulla- tai taulukolla, mutta vievät vähemmän tilaa – joskus jopa vähemmän kuin merkkijono T itse, jos T:ssä on paljon toistuvuutta. Nämä perustuvat usein niin sanottuun Burrows-Wheeler-muunnokseen, jota käytetään muutenkin datan pakkaamiseen. Pakattujen tietorakenteiden parhaat palat ovat helposti saatavilla Githubista **Simon Gogin** loistavasta sds-lite-kirjastosta, jossa on yli neljänkymmenen tutkimusartikkelin tietorakenteet ja algoritmit siistinä C++ template-kirjastona.

Toinen aktiivinen tutkimussuunta on suurpiirteinen haku, jossa etsitään täsmällisten vastaavuuksien sijasta *samankaltaisia* merkkijonoja, jotka muistuttavat toisiaan tarpeeksi jollain

kriteereillä. Esimerkiksi saatetaan hakea pareja osamerkkijonoja, jotka ovat voidaan muuttaa toisikseen muutamalla merkin lisäyksellä tai poistolla. Tämä on hyödyllistä esimerkiksi geenien hakemisessa tietokannasta, koska halutaan olla välittämättä pienistä eroavaisuuksista DNA:ssa tapahtuneiden mutaatioiden vuoksi. DNA-ketju tulkitaan merkkijonoksi, jonka aakkosto on DNA-molekyylin komponentit A, C, G ja T.

Jokapäiväisissä sovelluksissa haettavan merkkijonon S pituus on yleensä lyhyt, joten listauksen I kaltaiset naiviit algoritmit toimivat kohtuullisen nopeasti. Alaa eteenpäin ajaakin pääasiassa bioinformatiikassa yleiset valtavat merkkijono-ongelmat, joissa saatetaan vertailla keskenään miljardien merkkien pituisia merkkijonoja, jotka esittävät esimerkiksi DNA-ketjuja. Rahoitusta tutkimukselle löytyy runsaasti, koska tehokkaalla merkkijonohauulla on sovelluksia geneettisten sairauksien diagnosoinnissa ja potilaan genomille räätälöidyn lääkityksen määrittämisessä. Tehokkaat merkkijonoalgoritmit pelastavat ihmishenkiä jo nyt. 🐺

Pähkinöitä: pystytkö ratkomaan $O(n)$ -ajassa seuraavat ongelmat?

- Etsi kahdesta merkkijonosta pisin osamerkkijono, joka esiintyy molemmissa.
- Etsi useimmin esiintyvä osamerkkijono, jonka pituus on vähintään 10.
- Etsi pisin osamerkkijono, joka on palindromi.
- Etsi merkkijonon aakosjärjestyksessä pienin rotaatio.

Kirjallisuutta

- **Crochemore & Rytter:** *Jewels of Stringology*
- **Gusfield:** *Algorithms on Strings, Trees, and Sequences*

SURKEAT suomipelit

Teksti: Jukka O. Kauppinen

Kuvat: Jukka O. Kauppinen, Toni Kortelahti, Mobygames



Suomi on tänään pelimaailman kärkimaita, etenkin pienuuteensa nähden. Hyviä huippupelejä saapuu tasaiseen tahtiin sekä mobiiliin että isommille koneille, mutta ei kaikki voi aina olla silkkaa aurinkoa ja hymynaamoja. Kyllä Suomessa on tehty huonoja, karmeita ja öklöjäkin kaupallisia pelejä. Ja varmasti tehdään jatkossakin.



Huonon tai paskan pelin käsite on toki epämääräinen ja täynnä henkilökohtaisia tulkintoja. Kuka tykkää mistäkin. Hyvän pelin huonompi jatko-osa ei ole todennäköisesti paska, vain huonompi. Ja joskus jotkut laukovat tosissaan aivan uskomattomuuksia. Esimerkiksi Remedy'n Alan Wake on aika helkkarin kaukana paskapelistä.

Ilmaispelien ja indien maailmasta kakkaa toki löytää suunnattomasti, etenkin jos aikahaitaria venytetään tarpeeksi. Ysäriin suomipelimaailma oli täynnä kauheaa roskaa, jonka seulominen MikroBittiin arvosteltavaksi ja MBnet-purkkiin suuren yleisön ladattavaksi oli joskus kovaa hommaa. Sieltä löytyi paljon helmeileviä teoksia, jotka yritin nostaa esille. Kamalaa roskaa oli kuitenkin uskomattoman paljon, ja ne saivatkin vauhdikasta del-kyyttä. Oli toki kivaa, että Jonne-Petteri halusi tarjota uutta tekstiseikkailuaan julkaistavaksi, mutta...

Kaupallisten suomipelien haukkuminen on kuitenkin yllättävän vaikeaa, muun muassa siksi, että niiden taso on keskimäärin melkoisen korkea. Lisäksi pelejä julkaistaan niin kiivaaseen tahtiin, ettei niistä pysy enää kukaan perillä. Vaikka seassa on varmasti sattumiakin, niitä ei todennäköisesti enää edes huomaa. Ja kuka nyt tietoisesti huonoja pelejä etsii? Jos jonkin pelin sitten haukkuu pataluhaksi, niin sen tekijähän saattaa tulla joskus antamaan henkilökohtaista palautetta!

Tässä jutussa kulma on nimenomaan kaupallisesti julkaistuissa, vanhemmissa teoksissa. Vinkkejä uudemman ajan surkeuksista voi lähettää osoitteeseen jukka@skrolli.fi.



Avesoftin julkaisu oli tasoltaan keho ilmais- tai shareware-peli.

Crapiolan ytimessä

Ikivanhojen, lähes indiehenkisten räpellysten haukkuminen on kaksipiippuista hommaa. Jostainhan aina pitää aloittaa. Jos siinä sivussa pelistään saa rahaakin, niin aina parempi. Avesoftin julkaisemaa Amiga-räiskintä Bloody Afternoonia ei juuri historiankirjoista löydä, mutta sillä on yllättävää historiallista merkitystä.

Aivan kamalahan se on. Bloody on hirveä pelihalliräiskintä, jossa räimitään rakennusten ikkunoihin ilmestyviä mörkkejä, ennen kuin nämä ampuvat sinua, ja yritetään olla ampumatta vähemmän vihaisia naamoja. Äänet ovat upean hirveitä ja peli on sisällöllisesti, pelillisesti ja olemukseltaan niin surkea että hirtittää. Kamalaa ajatella, että tästä on keh-dattu pyytää rahaakin. Mutta eipä Avesoftikaan aikoinaan peliä kovin hyväksi noteerannut.

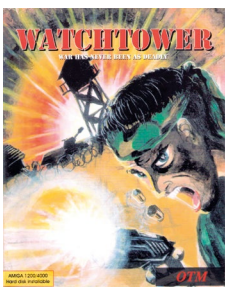
Avesoft-pomo **Jyrki Kummola** itse asiassa kommentoi aikanaan allekirjoittaneelle, että ”se on paras Basic-peli, jonka olen ikinä nähnyt” ja muisteli, että kun peliä tarjottiin brittiläiselle halpapelikustantajalle, niin ”torjunta oli tyyli”.

Bloody onkin jäänyt monien mieleen ”hirveintä skeidaa mitä Amigalla olen ostanut” -tasoiseksi klassikoksi. Mutta samalla se oli graafikko **Saku Lehtisen** ensimmäinen kaupallinen julkaisu. Myöhemmin Sakun kädenjälkeä on nähty muun muassa peleissä Max Payne, Max Payne 2, Alan Wake ja Quantum Break. Ikihirveästä gorejuhlasta urkeni siis hieno ura!

Bloody Afternoon (1989)

Tekijä/julkaisija: Avesoft Oy

Alusta: Amiga



Watchtower yhdisti kuluneita ideoita ja valjuun toteutukseen. Ei ihme, ettei maistunut.

Kankea Commando '96

Watchtower saapui Amiga 1200:lle koneen tähden ollessa jo vahvassa laskusuunnassa. No, eihän se mitenkään estänyt hyvien pelien tekemistä, etenkin kun AGA-koneiden kikkorja opittiin hyödyntämään paremmin loppua kohden. Mutta mikä avuksi, jos bitti ei vain suostu yhteistyöhön?

Yli vuosikymmenen takaista Commandoa vahvasti lainannut suomipeli soitti monia vääriä säveliä, ja bitti oli poikittain. Olihan siinä sentään jonkinlainen idea, kulunut mutta kuitenkin. Maailmaa kiertävä palkkasoturi rynnistää

ruudulla alhaalta ylöspäin suorittaen kuutta tehtävää eri ryhmittymien leivissä ja roimii vihuja seitsemällä eri aseella. Ylhäältä kuvatusta toiminnasta pystyi nauttimaan joko yksin tai samalla ruudulla kaverin kanssa.

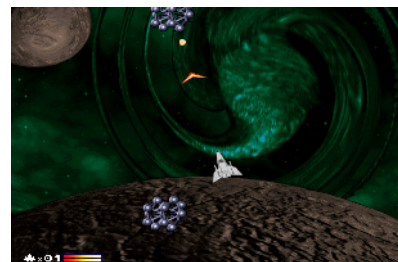
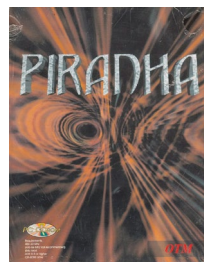
Tai nauttimaan ja nauttimaan. Watchtower oli tylsä ja turhauttava räiskintäpeli, joka oli AGA-grafiikastaan huolimatta valjun näköinen ja säväyttämätön peli. Konetuliaseiden pötpötykselle vetivät vertaa vain antiikkinen design, ilmaan haihtuvat ammuksat ja mukatekoälyllisesti aivokuolleet viholliset. Se oli yksinkertaisesti peli, joka ei inspiroinut ja muuttui pian monotoniseksi junnaamiseksi. Liian kliseistä, liian myöhään, liian tutulla reseptillä.

Pisteet sentään siitä, että CyberArts sai vielä '96 kasaan Amiga-pelin, jopa AGA-alustalle, ja sai sen kaupalliseen julkaisuunkin. Veikkaan silti, ettei tämä myynyt kovinkaan montaa kymmentä kappaletta. CyberArtsin esikoispeli jäi myös viimeiseksi. Paljoa paremmin ei mennyt julkaisijalakaan – OTM sai tiettävästi ulos tasan kolme peliä. Niistä kaksi suomalaisia.

Watchtower (1996)

Tekijä/julkaisija: CyberArts Ltd. / OTM Publications & Promotions Ltd.

Alusta: Amiga AGA



Tuntemattomiksi jääneiden tekijöiden Piranha kompastui pahasti Stardustin upeuteen.

Stardustin jäljillä

Asteroids-kloonit ovat olleet vuosituhanesta toiseen yksi videopelaamisen kulmakivistä ja tavallaan myös yksi suomalaisen peliteollisuuden juurista. Bloodhousen Amigalla 1993 synnyttämä Stardust-sarja kun on porskuttanut sittemmin Housemarquen kulmakivenä tarjoillen huikeaa pelattavaa etenkin nykypleikkareille. Mutta muutkin kokelivat lohkaroiden murskausta avaruudessa ja paljon kurjemmin tuloksin.

Vaikka syvälle hanuriin työnnettäviä indie- ja harraste-Asteroidseja piisaa, niin Piranha nousee omilla mittareillani kärkisijalle. Ehkä ennen kaikkea siksi, että siitä jäi suuhun niin paha maku. Amigan Stardust, 1994 Amiga AGA:lla ja CD32:lla sekä 1996 DOSilla nähty Super Stardust olivat upeita, liki täydellisiä Asteroids-jälkeläisiä. Sitten Piranha roiskii rapaa silmille ja seinille niin, että huudattaa.

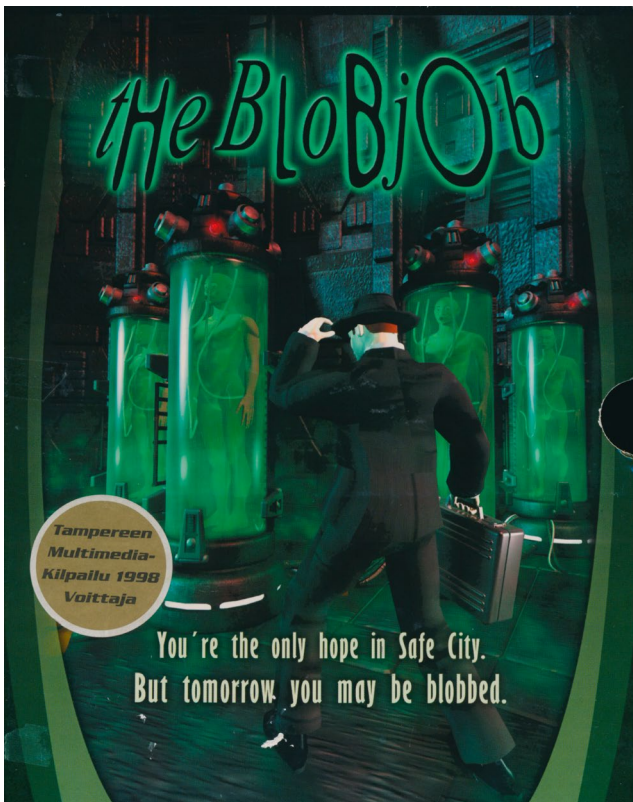
Piranhassa oli yrityksenpoikasta ja jotain uuttakin kuten monen pelaajan deathmatchaus ja räiskintäliigat, mutta samalla peli oli tyylitön ja tauhkainen. Demonilaumat piilivät yksityiskohdissa. Grafiikka oli periaatteessa koreaa, mutta myös vaisua ja puhuttelematonta. Musiikit eivät tylsyydessään säväyttäneet ja äänirutiinikin rahisi. Pahinta oli viilaamaton pelattavuus ja pelimoottorin kurjuus, sillä lohkarimäärän kasvaessa peli hidasteli ja nyki kelvottomasti. Asteroidsin henkeäkään ei oikein ollut tajuttu, sillä ruudulle

sydettiin hirveästi liikkuvaa kamaa ajattelemaan lainkaan, missä kohtaa haastavuuskäyrä muuttuu v*t*t*skäyräksi. Voinee väittää, että peli oli enemmänkin kasa yhteen heitettyjä rutiineja ja kierrätettyjä ideoita, mutta kukaan ei ollut miettinyt tai suunnitellut kokonaisuutta.

Ilmeisesti Piranhan tekijöillä oli demoskenetaustaa, mutta valitettavasti tekijätiimistä tiedetään vain muusikot. D-Designista ei ole jäänyt minkäänlaista nettijälkeä. Jos jollakulla on käsitystä kavereista, saa kärehtyä allekirjoittaneelle, sillä kehnoudesta huolimatta Piranhankin historiaa olisi kiva tallentaa. Myös Piranha oli brittiläisen OTM:n julkaisu. Firman kolmas, paras ja viimeinen julkaisu tuli muuten Italiasta.

Piranha (1996)

Tekijä/julkaisija: D-Designs / OTM Publications & Promotions Ltd.
Alusta: PC



Blobjob ansaitsi uudelleenjulkaisun Suomi100:n kunniaksi!

Ylinäyttely ja valistusta

The Blobjob on ideastaan lähtien hirmuisen hämmäntävä peli. Se on näet isoa sormea heristelevä valistuspelejä, joka valistaa nuoria muun muassa huumeista, liikenteen vaaratilanteista, turvaseksistä, viinasta ja sähköpaloista. Valistus-aspekti tosin yritettiin piilottaa oudon tieteistarinan taakse. Viisi synteettistä ihmistä on kadonnut matkallaan patentti-virastoon, ja nyt heidät pitäisi löytää.

Samalla Blobi on myös suttuinen multimediapeli, joka on hieno esimerkki ysärin CD-ROM-vallankumouksen kammalimmista puolista. Tuohon aikaan oli kuuminta hottia kuvata eläviä näyttelijöitä ja ympäri peleihin CD-ROM-levyltä toistettavaa videota. Niinpä Blobjobissa kuvattiin näyttelijöitä bluescreeniä vasten ja istutettiin heidät tietokoneaustoihin. Tähän päälle mausteeksi point'n'click-seikkailua ja puzzleja, niin johan luulisi käsissä olevan klassikon ainekset.

Vähemmän yllättäen lopputulos on kaamea ylinäytelty reiliikki, jossa elävä kuva pyörii ruudulla tikkuaskin kokoisena, kiitettävän niukoilla frameilla. Sillä jos näyteltyä pelielokuvaa kerran tehtiin, niin säästetään nyt varmuuden vuoksi levytilaa esittämällä ihmisiä muutaman kuvan mittaisilla looppaavilla pätkillä. Kaupan päälle seikkailun ongelmat ovat lähipuutojuttuja eikä läpipeluuseen mene puolta tuntia pidempään.

Muistettakoon kuitenkin, että tämä on aikansa teos. Arvostan kovasti sitä, että tällainen on pystytty aikoinaan Suomessa tekemään. Kansanvalistusaspektinkin selittynee sillä, että tekijät ovat varmaan alkujaan tehneet tietesseikkailua, mutta rahoittaja on halunnut mukaan myös opettavaisia elementtejä. Ei ehkä aivan ideaalinen tapa tehdä peliä.

Blobjob muuten voitti Mindtrek-multimediakilpailun 1998. Tämä on siis palkittu teos! Ei ole moni suomalaispeli näin hienoja pokaaleja kerännyt, joten sen haukkuminen on siis loppujen lopuksi vain kateellisten panettelua.

The Blobjob (1998)

Tekijä/julkaisija: Intelligame Oy, Detonium Interactive, Sarajärvi & Hellén Oy / Sampo Oy
Alusta: PC

Snaken jatko-osat

Taneli Armannon Nokian kännyköille tekemästä Snake-matopelistä kasvoi käsite ja ensimmäinen suosittu mobiilipeli. Nokia yritti ratsastaa myöhemminkin matoilulla, mutta kovin hyvin siinä ei käynyt. Yksinkertainen oli kaunista, kikkailu ei.

Nokia teetätti Snakelle monta jatko-osaa, jotka olivat toki kauniimpia ja värikkäämpiä – mutta myös pettymyksiä. N-Gagen Snakes pursusi värejä ja erilaisia pelitiloja, mutta kompastui kikkailuun. Oli se silti yksi N-Gagen paremmista peleistä. Useisiin Nokian malleihin valmiiksi asennettu Snake III panosti piirroselokuvamaiseen, söpöön tyyliin isometrisellä 3D-henkisellä kuvakulmalla. Peli oli visuaalisesti hirmu



Antaa matopelin maata mullaksi, turha sitä on enää kierrättää edes uudessa 3310:ssä.

hieno, ja idea laajoista, ruutua suuremmista kentistä ja vie-rivästä ruudusta oli oivallinen. En silti lämmennyt pelille yhtään, sillä loppujen lopuksi näitä matopelejä oltiin kuitenkin nähty monessa muodossa aiemminkin. Olihan se söpö, mutta niin olivat kloonitkin. Snakes Subsonic taasen oli teknisesti upea, mutta myös suorastaan suttuinen, epäselvä ja epäpelattava. Kaikessa tronimaisuudessaan peli oli pienellä ruudulla liian iso ja korkealla kurrottava.

Jatko-osien onnistumisen tasosta kertookin paljon se, ettei niitä muisteta tänään lainkaan. Alkuperäisen Snaken nerokas yksinkertaisuus sen sijaan toimii yhä edelleen.

Snakes (2005) / Snake III (2005) / Snakes Subsonic (2008)

Tekijä/julkaisija: useita / Nokia

Alusta: Nokian puhelimet

Kunniamaininnat



Rakastettukin peli voi olla toisille hirvitys.

80-luvun kuuluisimpiin lukeutuvat suomipelit Painterboy ja Uuno Turhapuro ovat pelejä, jotka muistetaan ja joita yhtä aikaa sekä rakastetaan että vihataan. Ne eivät ole suinkaan huonoja pelejä, vaan pikemminkin erinomaisia aikansa peilejä. Molemmissa oli mittaansa nähden korkeat tuotantoarvot, ja niitä onkin pelattu raivolla ja tunteella. Monesti ennen kaikkea raivolla, sillä molemmat pelit ovat hillittömän vaikeita. Pelaajan tunteita ei suuremmin ajateltu, eikä pelitestaustakaan juuri harrastettu. Tekijät viilasivat vaikeustason oman osaamisensa mukaan, ja kun suuri yleisö päästettiin pelien kimppeeseen, niin kyllä siinä tikut lentelivät seinään.

Onkin jamppoja, joille jompikumpi tai kumpikin peli edustaa paskuuden huipentumaa. Se saattaa henkilökohtaisella tasolla pitääkin paikkansa. Ehkä niiden yliampuvaa vaikeustasoa voi pitää tänään huonona designina. Mutta 1980-luvulla moni asia oli toisin. Ne ovat suomiklassikoita, joita ei tänään nautinnokseen pystyisi pelaamaan, mutta eivät ne mitään pohjamutaakaan ole.

Nykyajan hiteistä mieliä jakaa erityisesti mobiilipeli Hill Climb Racing, josta kauneimmat kiertelyt toteavat ”ei niin kova kuin maineensa”. Sen jatko-osa on herättänyt hieman erilaisia tunteita, sillä ainakin kyselyjen ja jutustelujen perusteella moni pitää sitä pelimekaniikoiltaan selvästi edeltäjänsä huonompana.

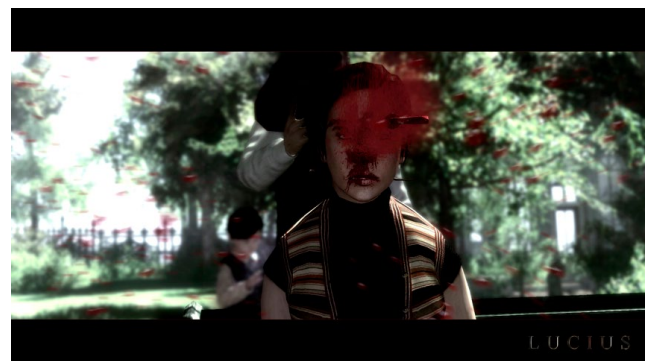
Lähes mutta melkein

Kuten palstan alussa mietiskelin, suomalaispelien kritisointi on vaikeaa. Joskus jopa ikävää. Esimerkiksi Shiver Gamesin Lucius-seikkailut ovat pelejä, joita haluaisi kehua ja ylistää.

Niissä on ideaa, urheaa yritystä ja sinnikkyyttä. Pikkuriikkinen indiestudio tavoitteli niillä korkealle, mutta aika, raha ja resurssit eivät riittäneet lupaavan idean viemiseen aivan perille saakka.

Kohtasin aikoinaan Luciusin tekijän pelimessuilla, haastattelin häntä, tein positiivisen ennakkotutun ja jäin pitämään peukkuja. Ikävä kyllä peli ei loppumetreillä tavoittanut lupaavan asetelmansa potentiaalia. Siinä oli ideaa ja kekseliäisyyttä, sillä harvemminpa peleissä pääsee touhuamaan Saatanan poikana ja aiheuttamaan maan päälle kaaosta ja kuolemaa. Seikkailua tehtiin hyvin kotikutoisesti, itse kehitetyillä työkaluilla ”tein itse ja säästin” -tyylisellä motion capture -tekniikalla, paljolti yhden ainoan ihmisen voimin. Se vain ei riittänyt. Pelisuunnittelun ongelmakohdat, bugit, kömpelöt kontrollit, suoraviivaisesti toteutetut kaaostehtävät ja paikoittain epäselvä ”mitä se pelintekijä nyt oikein haluaa minun tekevän” -logiikka kaatoivat seikkailun.

En nosta Luciusia esille siksi, että se itsessään olisi kummoinenkaan katastrofi, vaan siksi, että se on osuva esimerkki suomipeleistä, jotka eivät aivan onnistuneet. Monien kanssasisartensa ja -veljiensä kanssa se on yksi monista teoksista, jotka jäivät yrityksestä huolimatta tasolle ”tämä voisi olla kiva peli, todella hyväkin, jos vain...” 🐉



Saatanan pojan murhaseikkailussa oli ideaa.





Ilmaisia lautapelejä verkossa – mutta mihin hintaan?

Lautapelit ovat nykyään hirvittävän suosittu harrastus, jolla riittää ystäviä ympäri maailman. Teknologian kehittyessä kuvioihin on ilmestynyt myös aivan uusi ilmiö: lautapelien piratisointi. Mistä on oikein kyse, ja onko asia sittenkään aivan niin mustavalkoinen?

Teksti: Miikka Lehtonen

Kuvat: Miikka Lehtonen, Emma Kantanen, eri pelivalmistajat

Kaikki alkoi – kuinkas muuten – Saksasta. Lautapelaaminen oli jo vuosien ajan ollut lasten, vanhempien ja armottomien turbonörttien hommaa, eli aika pientä marginaalipuuhastelua, paitsi tietenkin Saksassa. Siellä lautapelit olivat kohdallaisen suosittuja vielä vuonna 1995, jolloin *Settlers of Catan* räjähti suosioon maailmanlaajuisesti.

Peli ei suinkaan jäänyt yksinäiseksi ihmeeksi, vaan sai nopeasti seuraa ja kilpailijoita kaikkialta maailmasta. Ihmiset huomasivat, että lautapeleissä riitti ideoita. Kiinnostuksen ja yleisö-
määrien noustessa myös tuotantobudjetit kasvoivat ja yhä kunnianhimoisemmat pelit saivat vihreää valoa julkaisijoilta. Nykyään, reilut 20 vuotta myöhemmin, lautapelit ovat jo valtava bisnes. Niitä pelaavat kaikenlaiset ihmiset kodeissa, työpaikoilla, baareissa ja ties missä muualla.

Eräs lautapelaamisen suurista valteista on myös erällä tavalla sen suuri

rajoitin. Vaikka poikkeuksiakin löytyy, valtaosa lautapeleistä on sosiaalisia kokemuksia, joihin tarvitaan useita pelaajia. Kuten jokainen useampia ihmisiä vaativia harrastuksia järjestänyt tietää, tämä ei ole aina aivan hirveän helppoa. Varsinkin aikuisilla aikataulut ovat usein sellaisia, että on hankala saada puoli tusinaa ihmistä saman pöydän ääreen illanviettoa varten.

Ratkaisu ongelmaan on toki ollut olemassa niin kauan kuin tietoverkko-kin, sillä jo ensimmäisistä tietokoneiden väliset yhteydet mahdollistavista ratkaisuista lähtien ihmiset ovat pelanneet niillä lautapelejä. Shakit, tammets ja muut perinteiset pelit ovat saaneet rinnalleen pelaajien ja harrastajien kehittämää työkaluja, joilla on voinut pelata jos jonkinlaisia pelejä verkossa.

Eräs verkkopelaamisen pioneereista oli VASSAL, Java-pohjainen pääteohjelma, joka suunniteltiin raskaiden sota- ja miniatyyripelien pelaamiseen verkossa. Ohjelma on nauttinut tasais-

ta suosiota jo useiden vuosien ajan, joskin lähinnä vain omista piireissään. Kankea ja karu ohjelma ei ole hirveän helposti lähestyttävä, eivätkä sen simuloimat pelitkään ole niitä kaikkein suosituimpia.

Samaan kategoriaan kuuluu myös *FumBBL*, jonka avulla Games Workshopin mainiosta fantasiafutispeleille *Blood Bowl*ista pitävät ovat pelailleet omia sarjojaan ja liigojaan jo muutamana vuosikymmenen ajan.

Muitakin vastaavia ohjelmia on ollut tarjolla jo pitkään ja mitä ihmeellisempiin tarpeisiin, mutta niissä kaikissa on ollut samat ongelmat – jos niitä sellaisina nyt haluaa pitää. Ohjelmat eivät olleet kovinkaan helppokäyttöisiä saati näyttävän näköisiä, eikä niillä ollut paljoakaan käyttäjiä.

Virtuaalipöydän ritarit

Lautapelien nostaessa profiiliaan ihmisten olohuoneissa ja pöydillä PC-pelaamisen maailmassa koettiin

samanlaisia suosionnousun hetkiä. Valve-yhtiön Steam vahvisti jatkuvasti otettaan pelaavan maailman suurimpana verkkokauppana. Digitaaliseen peli- ja ohjelmistojakeluun keskittyvä Steam tarjoaa kehittäjille suuren yleisön ja pelaajille helpon pääsyn toinen toistaan ihmeellisempiin peleihin.

Eräs Steamin jättimäisen kasvun tekijöistä on ollut Valven pyrkimys lisätä siihen kiinnostavia ja hyödyllisiä ominaisuuksia. Meidän aiheemme kannalta keskeinen ominaisuus on Steam Workshop, hämmäntävän helppokäyttöinen ja suosittu tapa kehittää, jakaa ja hallinnoida erilaisia modeja, fanien tekemiä lisäosia ja muita kikkkeitä Steamissa myytäviin peleihin.

Moni peli on käyttänyt Steam Workshopia ihailtavan kekseliäillä tavoilla, mutta harva niin hyvin kuin *Tabletop Simulator*, jota ei voisi olla edes olemassa ilman Workshopia. *Tabletop Simulator* on kesällä 2015 julkaistu ohjelma, jonka koko pyrkimys on tarjota graafisesti näyttävä ja täysin kolmiulotteinen virtuaalitala, jossa voi pelata erilaisia lautapelejä.

Kehittäjien vision mukaan *Tabletop Simulatorin* pyrkimyksenä oli tarjota työkalulaatikko, jonka avulla pelaajat voisivat pyörittää vaikka *Dungeons & Dragons* -pelejään virtuaalisesti aivan kuin he istuisivat saman pöydän ääressä. Pelaajat näkevät ruuduillaan virtuaalisen pöydän, jonka päällä kaikki komponentit ovat kuin oikeina fyysisinä versioinaan. Sen sijaan että pelinappia siirtäessään pelaajat klikkailisivat ikoneita ja ruutuja, he tarttuvat hiirellään kohteeseen ja siirtävät sitä virtuaalimaailmassa.

Voisi hyvin kuvitella, että tällainen miltei skeuomorfistinen pyrkimys toistaa oikean elämän rituaaleja ja elementtejä virtuaalisesti työntäisi pelaajia pois. Intuitiivisesti juuri ne yksinkertaiset ja nopeat hiirivetoiset käyttöliittymät tuntuisivat paremmalta

ratkaisulta, mutta toisin tuntuu käyneen: *Tabletop Simulatorista* tuli nopeasti suuri hitti, joka kipusi pikavauhtia Steamin myyntilistojen kärkeen ja pysyi siellä sitkeästi kuukaudesta toiseen.

Käyttöliittymä ei varmasti ole se ainoa tai edes suurin syy tähän, sillä *Tabletop Simulator* on tarkoitustaan varten erinomaisen hyvin suunniteltu paketti. Se tarjoaa työkalut kommunikointiin, peliseurauksen etsimiseen, skriptaustryökalut erilaisten sattumien ja noppien simulointiin ja niin edelleen.

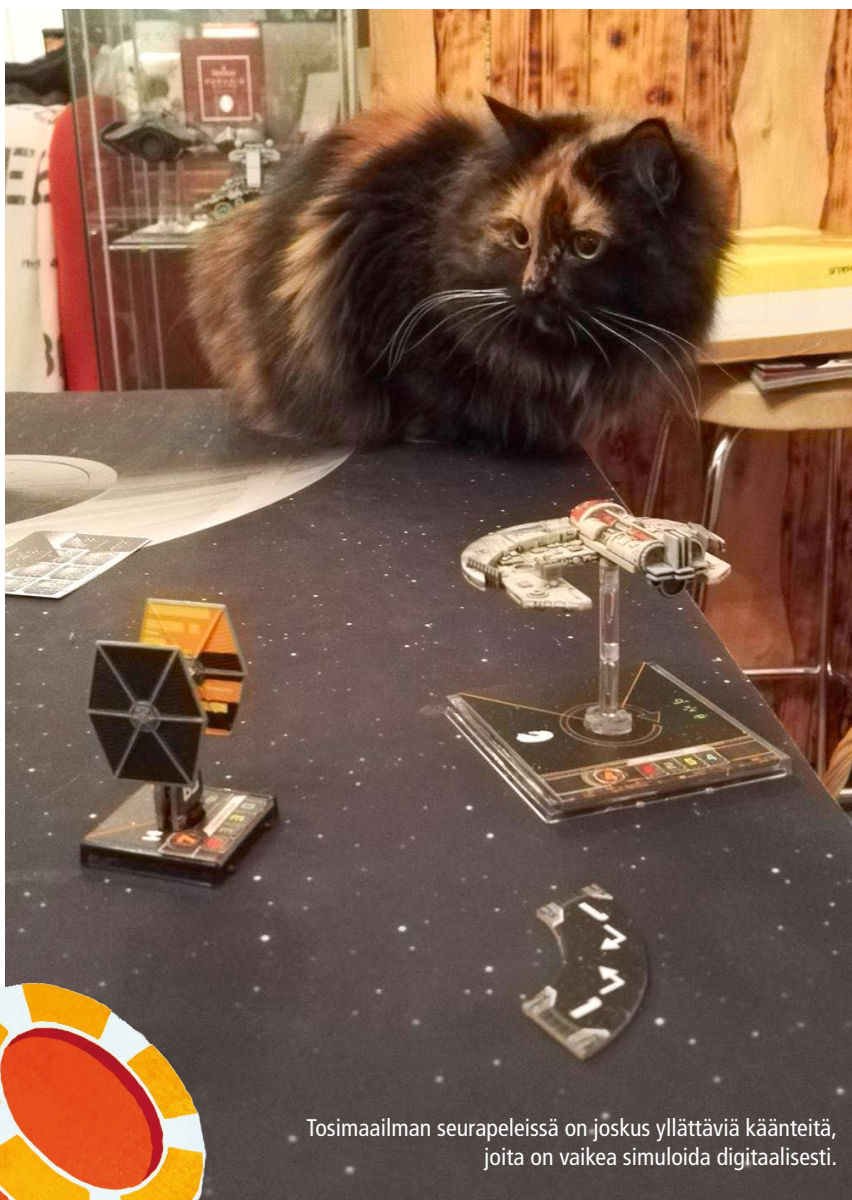
Niin, ja tietenkin myös työkalut, joilla pelaajat voivat itse tehdä omia lautapelejä. Jotenkin veikkaan, että juuri tämä on se suurin syy suosiolle.

50 euroa? Vai 0 euroa?

Lautapelaaminen ei ole halpa harrastus. Yksittäiset pelit maksavat halvim-

millaankin muutamia kymmeniä euroja keskihinnan ollessa käytännössä 50–60 euron kieppeillä. Kalliimmat, eli enemmän hienoja komponentteja sisältävät pelit, voivat kustantaa reilusti yli 100 euroa. Täten on selvää, että ihmiset ovat halunneet kopioida pelejä suunnilleen niin kauan kuin he ovat halunneet niitä yleensäkin pelata.

Perinteisesti tämä ei ole ollut helppoa, sillä siinä missä digitaalisen ohjelman voi kopioida helposti, hirmuinen pino fyysisiä komponentteja vaatii jo enemmän vaivaa, eikä jälki ole kuitenkaan läheskään samaa tasoa kuin alkuperäisessä. Täten lautapelipiratismi ei olekaan koskaan ollut mikään suunnaton juttu, mitä nyt muutamat huru-ukot vähän ovat valokopioineet *Magic: the Gathering* -korttejaan.



Tosimaailman seurapeleissä on joskus yllättäviä käännteitä, joita on vaikea simuloida digitaalisesti.





Tabletop Simulatorin avulla pelaajat ovat tehneet myös omia erikoisuuksiaan, kuten tämän Bang!-korttipelin World of Warcraft-version.

Tabletop Simulator kuitenkin muutti tilanteen totaalisesti. Yllättäen korkearesoluutioiset skannit pelin korteista ja pelilaudasta, sekä 3D-mallinnusohjelmilla tehdyt versiot sen fyysisistä pelimerkeistä riittivät tuottamaan kokemuksen, joka on hyvin lähellä alkuperäistä. Tabletop Simulatorin voimakkaan skriptauskielen ansiosta peleihin on myös helppo toteuttaa yllättävän monipuolisia mekanismeja: pusseja, joista voi nostaa satunnaisia pelimerkkejä, itsestään sekoittuvia pakkoja ja jopa pelin logiikkaa valvovia ja seuraavia ”pelinjohtajia”.

Myös oikeiden fyysisten pelien kehittäjät ovat huomanneet tämän. Tabletop Simulatoriin onkin kaupan virallisia lisäsisältöpaketteja, jollaisena voi vaikka 8,99 euron hinnalla ostaa virtuaaliversion suosituista *Zombicide*-pelistä. Oikea versio kustantaa lähemmäs 100 euroa, joten paperilla tarjous kuulostaa ihan hyvältä. Pelikokemushan kuitenkin on ainakin mekaanisesti hyvin lähellä oikeaa, vaikka oikeiden pelien sisältämien artefaktien käpistelystä saatavaa mielihyvää ei pääsekään kokemaan.

Käytännössä nämä lisäsisällöt kuitenkin pölyttyvät virtuaalilyhyille, sillä tarjolla on vielä halvempi versio: fanien tekemä Steam Workshop -modi, joka tarjoaa kaiken saman – ja usein vielä enemmänkin – ilmaiseksi.

Ja nyt olemmekin harmailla vesillä.



Koska laatua ei mitenkään kontrolloida, to-della hienojen ja laadukkaiden pelipakettien ohella tarjolla on myös vähemmän komeaa jälkeä, kuten tässä Blood Bowl -pelissä.

Kolikolla on kaksi puoliskoa

Hetkinen, ”harmailla vesillä”? Eikös piratismi kuitenkin ole yksiselitteisesti väärin? Tästä voidaan olla monta mieltä, mutta itse sanoisin, että tilanne ei ole aivan niin helppo. En toki pidä sitä mitenkään ylistämisen arvoisena juttuna, että netissä levitetään suoraan sanoen piraattiversioita kaupallisista lautapeleistä, mutta asiaan liittyy muitakin näkökulmia.

Niin kliseiseltä kuin se kuulostaa, kokeilumahdollisuus on todellinen houkutin monelle. Lautapelit ovat kalliita, ja toisin kuin videopeleistä, niistä ei käytännössä ole saatavilla demoja. Jos haluaa kokeilla lautapeliä ennen ostopäätöstä, täytyy joko toivoa, että joku kaveri jo omistaa sen, tai tunkea mukaan demotilaisuuteen pelikaupassa tai messuilla.

Tabletop Simulatorin kautta kynnyks kokeilla pelejä on kuitenkin olemattoman pieni ja se kutistuu jatkuvasti. Käytännössä kaikkien lautapeliensä säännöt on saatavilla netistä digitaalisina latauksina, ja lisäksi pelaaminen on muutenkin muuttamassa muotoaan.

Eräs lautapelaamisen suurista trendeistä ovat niin sanotut hybridipelit, eli pelit, joissa fyysinen lautapeli ja jonkinlainen digitaalinen ohjelma, kuten älypuhelinapplikaatio, yhdistyvät. Joissain peleissä nämä hybridi-



Eldritch Horrorin kaltaiset pelit sisältävät niin paljon komponentteja, että keittiön pöydällä on oikeasti ahdasta. Virtuaalimaailmassa tilaa riittää.

ominaisuudet vaikka toimivat ikään kuin tekoälyn vastustajana: ne opastavat pelaajalle, miten lauta tulee asettaa ja sitten ohjaavat lautaa sen pelatessa pelaajaa – tai yksittäistä pelaajaa – vastaan. Olen itsekin näin kokeillut muutamaa 120 euron lautapeliä ja huomannut, että vaikka ne olivat ihan kivoja, ne tuskin olisivat 120 euron edestä kivoja.

Toinen pohtimisen arvoinen kysymys on virtuaalisuus. Minun lautapelikaverini asuvat pitkin Suomea, Eurooppaa ja maailmaa. Näen useimpia heistä muutaman kerran vuodessa, joitain vielä harvemmin. Kun tapaamme, kaivamme toki esiin fyysiset versiot suosikkipeleistämme tai *X-Wing Miniatures* -figupelin laivastomme ja pelaamme keittiön pöydällä. Kun emme näin pysty tekemään, en ainakaan itse tunne syyllisyyttä, jos lataammekin netistä jonkun tekemän modin, joka sisältää pelin komponentit uskottavina virtuaaliversiona. Toki



Kolme versiota samasta pelistä. Lords of Waterdeep oikeana lautapelinä, Tabletop Simulator -versiona ja virallisena iPad-pelinä.

vastuu rehellisyydestä jää itse pelaajan vastuulle: lataatko pelejä, joita et itse fyysisesti omista?

Voidaan pohtia myös pieniä markkina-alueita. Lautapelaaminen on globaali harrastus, ja sillä on ystäviä kaikkialla maailmassa – myös pienillä kielialueilla, joissa englantia tai saksaa, lautapelaamisen kahta *lingua francaa*, ei välttämättä kamalasti puhuta tai ymmärretä. Koska lautapelin fyysinen tuottaminen on hirmuisen kallista puuhaa, nämä pienemmät markkina-alueet jäävät ilman pelejä sekä kielellisesti että usein fyysisesti. Kun markkina-alue on pieni, pelejä ei lokalisoita. Kun tarpeeksi moni ei ymmärrä englantia riittävän hyvin, pelejä ei kannata myöskään tuoda maahan englanninkielisinä. Täten harrastajien ainoa vaihtoehto on tilata pelejä nettikaupoista, jolloin valuuttakurssit, toimituskulut ja tullit nostavat hinnat tähtitieteellisiksi.

Tabletop Simulatorin latauslistojen kärjessä koreilevatkin jatkuvasti fanien tekemät lokalisoinnit peleistä mitä ihmeellisemmille kielille. Harrastajilla riittää aikaa ja intoa kääntää pelien komponentit uusille kielille, eikä jake-lukaan kustanna mitään. Täten kynys tukea pieniä markkina-alueitakin kutistuu. Optimisti voisi myös ajatella, että ehkä näin tekemällä voidaan lähettää pelien julkaisijoille viesti siitä, että pienilläkin alueilla kysyntää riittää.

Virtuaalisille lautapeleille löytyy siis paljon muitakin käyttökohteita ja perusteluja kuin halu piratisoida kalliita pelejä. Olisi tietenkin älyllisesti epärehellistä sivuuttaa tätä näkökulmaa kokonaan. Varmasti jokaista demopeliä tai lokalisoitua versiota etsivää pelaajaa kohti *Tabletop Simulatorilla* on useita pelaajia, jotka vain haluavat ilmaisen lounaan. Tilanne ei ole siis helppo.

Mihin tästä?

Tulevaisuus on mielenkiintoinen. Lautapelit ovat tuskin katoamassa mihinkään, vaan ne pikemminkin vankistavat koko ajan asemaansa hyväksyttävänä ajanvietteen muotona. Myöskään kerran pullostaa päästettyä henkeä ei saa tungettua sinne takaisin. Vaikka *Tabletop Simulator* poistettaisiin myynnistä, joku muu vain tekisi saman jutun uudelleen. Toden-

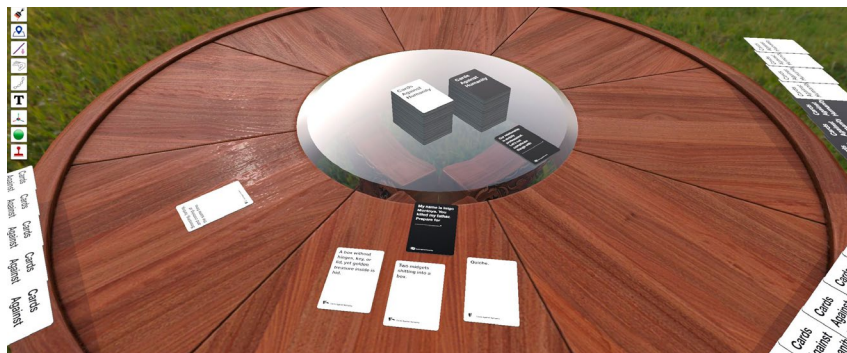
näköisesti vielä hämärämmin ja piratismikeskeisemmin.

Niinpä tilanne on pelaajien kannalta herkullinen, lautapelivalmistajien kannalta haastava. Moni firma varmasti pelkää menettävänsä myyntejään digitaalisille pelikopioille, ja moni muu haluaisi osaksi näistä uusista digirahoista. Vaan miten?

Vaihtoehtoja on useita. Moni firma on jo nyt ryhtynyt tuottamaan digitaalisia versioita peleistään ja ne ovatkin saavuttaneet suosiota. Lautapelit ovat usein algoritmisesti yksinkertaisia – ainakin mekaniikoiltaan – joten niiden kääntäminen digiversioiksi on kohtalaisen yksinkertaista puuhaa. Kysyntää riittää, ja miksipä ei? Nätti ja hyvin toimiva versio suosikkilautapeleistä, jota voi pelata verkon yli vaikka junassa tai hotellihuoneessa, on houkutteleva vaihtoehto.

Mutta mitä jos pelivalmistajat menisivät vielä askeleen pitemmälle? Mitä jos esimerkiksi Asmodee, pelimaailman suuri kolossi, tekisi oman *Tabletop Simulatorinsa*? *Asmodee Simulatorissa* pelaajat voisivat joko ostaa digiversiot peleistä pikkurahalla (fyysisiin peleihin verrattuna), tai fyysisen pelin mukana tulevan QR-koodin skannaamalla todistaa omistavansa pelin laillisen version ja saisivat näin oikeudet sen pelaamiseen kavereiden kanssa netissä.

Ongelma ei ole helppo, sillä pelaajia voi olla enää hankala saada siirtymään, varsinkaan maksullisten vaihtoehtojen pariin. Lautapelivalmistajien tappio on toisaalta pelaajien voitto, sillä lautapelin pelaaminen verkossa ei ole koskaan ollut näin helppoa tai mukavaa, ainakin jos ei anna moraalisten kysymysten painaa liikaa. 🐘



Cards Against Humanity on yksi parhaista ikinä koskaan tehdyistä seurapeleistä, jonka naamakkaissosiaalista puolta on mahdotonta toistaa parhaassakaan digiversiossa.

Myös miniatyyripelaajille

Jos lautapelit ovat hintavaa puuhaa, niin ne ovat lasten leikkiä verrattuna satojen eurojen panostuksia vaativiin miniatyyripeleihin. Ei olekaan yllätys, että juuri niiden mallintaminen on ollut hyvin suosittua puuhaa. Esimerkiksi megasuositun *X-Wing Miniatures* on niin hyvin tuettu *Tabletop Simulatorissa*, että kaikkien pelin alusten lisäksi siihen on saatavilla skriptattuja apuvälineitä, joiden avulla alusten lentoratojen mallintaminen ja muut pelipöydältä tutut käsitteet onnistuvat hyvin helposti.

Tabletop Simulator pähkinäkuoressa

Kehittäjä: Berserk Games

Hinta: 19,99 euroa

Laillisia, lisensoituja pelipaketteja:

28 erilaista nimikettä.

Fanien tekemiä pelipaketteja: yli 6000.

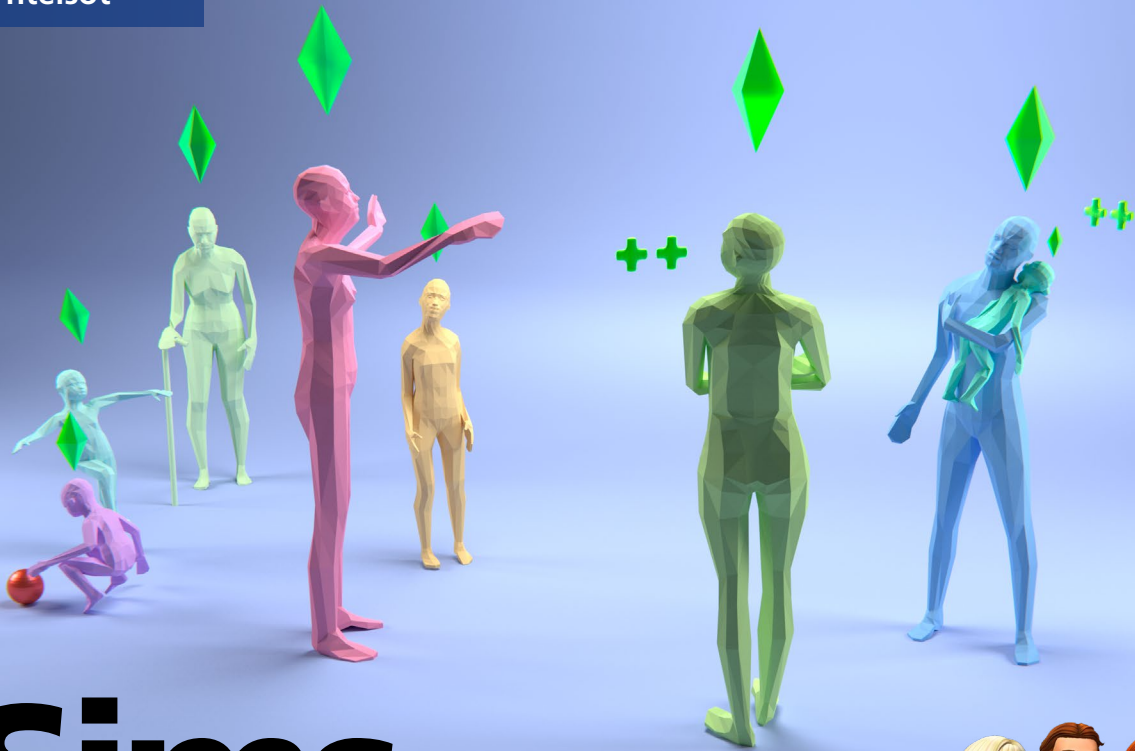
Aktiivisia pelaajia: parhaimmillaan yli 7000 samanaikaista pelaajaa.

Lisätietoja: berserk-games.com/tabletop-simulator



Immersiion seuraava askel

Tabletop Simulator tukee myös SteamVR-rajapintaa, eli esimerkiksi HTC Vive -lasit omistava saa todella immersivisen kokemuksen. Pikaisen testauksen perusteella VR-versio toimii oikein mainiosti. Tilan tuntu on hyvä ja pelaaminen on entistä mukavampaa, kun näppäinoikopolkujen sijaan voi liikkua fyysisesti pelitilassa ja manipuloida esineitä käsillään ohjainten avulla. Jää nähtäväksi, miten paljon vastaavat virtuaalilautapelikokemukset tulevat tulevaisuudessa yleistymään ja kehittymään. Milloin näemme virtuaalimaailmassa myös 3D-versiot kavereistamme?



Sims

Pelin eloisa ja erikoinen yhteisö

Teksti: Milsa

Kuvat: Toni Kortelahti, EA, Radosims, Radola

Sims-pelien ympärille on kasvanut Suomessa omanlaisensa aktiivinen ja tekemisen iloa pursuava yhteisö, jossa osataan luoda ja kannustaa.

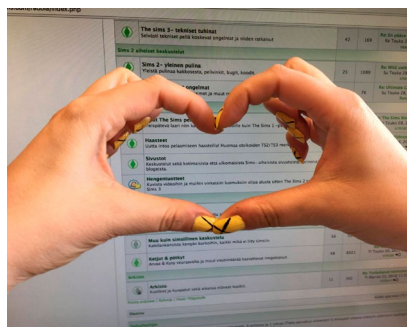
Maanmainio elämysimulaatiopeli *The Sims* syntyi helikopterisimulaation lapsipuolena, ja sen ensimmäinen valmis esine oli vessanpytty. Ensin sitä ei edes haluttu julkaista, mutta sittemmin sarjan eri osat ja laajennukset ovat myyneet satoja miljoonia kappaleita ja hurmanneet ruutujen ääreen nekin, joita muut pelit eivät inspiroi.

Vaikka *Sims-pääpeleissä* ei ole edes moninpeliä, niin niiden ympärille on tästä huolimatta kasvanut yhteisöjä, jotka toimivat niin pelin kautta kuin sen ulkopuolella. Samalla se on tietokonepeli, jonka parissa viihdytään lapsesta mummoon, ja joka yhdistää ihmisiä kautta maan. Se viitoittaa teitä opiskelupaikkoihin ja saa harrasta-

jansa opiskelemaan kuvankäsittelyn ja 3D-mallintamisen saloja.

Juoneton koukku – siis mitä?

Kuinka peli, jonka jännittävänä juonettomana juonena on luoda ja ohjastaa pieniä ihmishahmoja, simejä, läpi näiden elämän, on saanut aikaan tämän kaiken? Suomessa on ollut ensimmäisen *Simsin* julkaisusta saakka suuria *Sims-yhteisöjä*. Niistä suurin ja vanhin, *Radosims* ja sen keskustelualue *Radola*, pyöri edelleen, vaikka tietokantaharakiri on syönyt sivuston ja kaiken sinne luodun materiaalin moneen kertaan. *Sims-yhteisö* on kuitenkin jaksanut kasvattaa foorumin aina vain uudelleen nolasta uuteen kukoistukseen.



Verkkoyhteisö vetää edelleen puoleensa myös uusia kasvoja vanhojen, jo toista vuosikymmentä pelin parissa viettäneiden kavereiksi. Ikähaitari venyy ala-astelaisista aikaihmiisiin. Syitä on monia. Erittäin edesauttava asia on se, että *Simsissä* ei ole koskaan ollut yhtä suurta juonta, päämäärää tai tavoitetta, saati yhtä ainoaa oikeaa tapaa pelata. Ennalta määrättyjen kaavojen sijaan *Sims* on päivastoin taittunut kiltisti jos jonkinlaiseen pelityyliin. Pelaajat voivat asettaa itselleen omat tavoitteensa ja pelityylinsä, halusi sitten keskittyä vaikkapa rakentamiseen, hahmojen luontiin, niiden mielikuvitukselliseen listimiseen tai kaikenkattavaan kustomointiin. *Simsit* ovat aina tarjonneet alustan luovuudelle, ja jos jotain ei ole voitu tehdä valmiil-

la työkaluilla ja alustalla, pelaajat ovat tarttuneet itse toimeen.

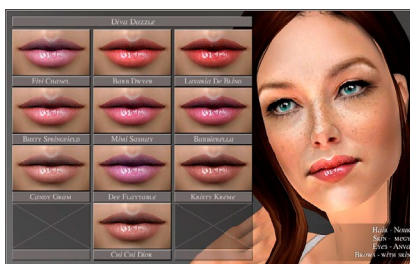
Grillataan vauvoja

Simsin pelaajat ovat luoneet peliin paitsi loppumattoman kavalkadin uusia vaatteita, meikkejä, hiuksia ja asusteita makuun kuin makuun. Samoin on loihdittu tuhannen Ikean verran uusia huonekaluja, kilometreittäin tapetteja, lattiaita ja muita kosmeettisia muutoksia naapurustoista rakennuksiin saakka.

Alusta saakka peliin on räätälöity myös modauksia, jotka muuttavat pelimekaniikkaa joko hillitysti sopivaan suuntaan tai rankemmin täysin uusiin suuntiin. Se mihin pelivalmistaja luomisen lopetti, on vain viiva, jolta pelaajat aloittivat.

Päivänvalon ovat nähneet paitsi pökerryttävän taidokkaat virtuaaliversiot huippumerkkien innoittamista vaatteista ja taidejäljennöksistä, samoin kuin vauvagrillin, orgiamaton ja teinraskaus-hackin kaltaiset keksinnöt. Voi olla, että ainakaan viimeksi mainitut eivät ole sitä, mitä pelivalmistaja olisi ajatellut saati toivonut pelaajien luovan. Mutta kuten aina, ihminen keksii alta aikayksikön tapoja, joilla nähdään enemmän paljasta pintaa, ja miten monin tavoin tällä pinnalla voidaan leikkiä.

Loputtomassa luomiskavalkadissa on ollut tilaa niin nahkapöksyille kuin hattarahiuksille, minkä ansiosta jokainen kotipelaaja on taatusti löytänyt jotain mieleistään. Ja jos ei ole löytänyt, niin on voinut tehdä sen itse tai voinut pyytää jotakuta mestarillista tekijää laatimaan sellaisen toiveiden pohjalta.



Brunon luomat tekstuurit mahdollistavat simien upean kustomoinnin esimerkiksi valokuvatärintöiden varten.

Eräs arvostetuimmista oman sisälön tekijöistä on suomalainen tekstuuritaitelija **Bruno**. Kansainvälisesti tunnettu taituri on niittänyt mainetta julkaisemalla äärettömän yksityiskohtaisia ja pökerryttävän hienostuneita meikkejä, ja hänen suosituimpia luomuksiaan on ladattu maailman suurimmalta *The Sims* -lataussivustolta, *Mod The Simsiltä*, yli satatuhatta kertaa. Brunon meikit ovat erityisen suosittuja Sims-aiheisissa tarinoissa ja videoissa.

Luomiseen on hyvä hurahda, sillä pelisarjan alkuvaiheessa Maxis-pelistödio teki etenkin tekstuurien luomisen varsin helpoksi ilmaisella Body Shop -ohjelmalla. Jostain kumman syystä, johon kenties liittyy rapeaksi grillattuja vauvoja, EA ei ole kuitenkaan tukenut pelaajien luovuuden kukintaa *Sims 3*- ja *Sims 4* -aikakaudella entisellä intensiteetillä. Tämä ei ole silti estänyt uusien luomusten tekemistä tai julkaisua.

Yhteisön yhdistämät

Loputtoman uuden ja uhkean materiaalin tulva, jolla pelistään voi boostata juuri toivomansa virtuaalinukkekodin, on ollut rakettipolttoainetta valtavien



Sims-modit mahdollistavat mitä oudoimpia juttuja.

Sims pähkinänkuoressa

Pelisuunnittelija **Will Wrightin** historia yltää vuonna 1989 julkaistun *SimCityn* perimään, jonka myötä herra loi erilaisia elämä- ja strategiasimulaatioita. Yksittäisten hahmojen elämää seuraava ihmismimulaatio sai alkunsa erään kirjan tarjoaman konseptin kautta, mutta muuntui varsinaiseksi peliksi vain hiljalleen. Vaikka suunnittelutyö alkoi jo 1991, saatiin peli kauppoihin vasta yhdeksän vuotta myöhemmin. Sinä aikana Wright sai Maxisilla valmiiksi useita muita pelejä, joista esimerkiksi *SimCopterissa* luotiin simlish-kieli.

- SimCity (1989)
- SimFarm (1993)
- SimTown (1995)
- SimCopter (1996)
- The Sims (2000)
- The Sims Online (2002)
- The Sims Bustin' Out (2003)
- The Urbz: Sims in the City (2004)
- The Sims 2 (2004)
- The Sims Stories (2007-2008)
- MySims (2007)
- The Sims 3 (2009)
- The Sims Medieval (2011)
- The Sims Social (2011)
- The Sims 4 (2014)

verkkoyhteisöjen syntyyn. Se ei silti ole yksin syy siihen, että moni vaippojaan sotkeva ihmistäimi on saanut alkunsa juuri Sims-yhteisöjen kautta. Kun yhteen keräytyy tuhansia ihmisiä jakamaan yhteisen intohimon, intohimo leiskuu helposti myös muualla kuin tietokonehuoneessa.

Niin ihanaa kuin loputon fanisisältöjen plaraaminen onkin, niin todellinen yhteisön moottori ovat silti ne kansapelaajat, jotka kokoontuvat yhteen vaihtamaan ajatuksiaan ja kokemuksiaan.

Oven tähän keskusteluun on usein tarjonnut tarve tarkistaa se paras mahdollinen huijauskoodi tai selvittää, että kuinka kummassa se maailman mahdollinen huvila voidaan rakentaa valmiiksi. Mutta samoin kuin pelisisältöjen luojille itse peli on vasta lähtöviiva, niin myös *Sims*-keskusteluissa pelistä turiseminen on vasta ensimmäinen muusa, jota muut jutut sitten seuraavat. Viaton mesh-ongelman kysymi-



Miiteissä voi tapahtua mitä vain.

nen on saattanut syöstä kysyjän läpi satojen aiheiden ja tuhansien viestien johtaen aina livetapaamisiin ja uravaliintoihin saakka. Hupsista!

Ei vihaa, vaan tukea ja kannustusta

Sims-yhteisöjä on alusta asti leimannut se, etteivät keskusteluaiheet rajoitu pelkästään peliin, vaan kattavat myös sen tuhannen muuta asiaa. Olivatpa kyseessä sitten koulukuulumiset, pohdinnat vapaa-ajasta tai ihmissuhteet, niin niistä keskustellaan aivan yhtä varmasti kuin Vauva-foorumia trolataan. Tuskinpa kukaan osasi olettaa, että pelisarjan ystävät rakentavat aikaa ja vaivaa säästelemättä sivustoja ja foorumeja, jotka kutsuvat tuhannet kanssapelaajat puhumaan lemmikeistään, tulevaisuudensuunnitelmistaan tai ihan vaan fanittamaan lempibändiään. Mutta niin vain kävi.

Yhteisten kokemusten hankinta ei rajoitu pelkästään verkon kautta tapahtuneisiin keskusteluihin ja elämyksiin. *Sims*-harrastajat kohtaavat toisiaan myös naamakkain: tapaamisia järjestetään ahkerasti etenkin Turussa ja Helsingissä. Niiden aikana syödään, juodaan ja pidetään yleisesti todella kovaa ääntä sekä puhutaan syntyjä syviä. Miitit ovat monelle tärkeitä ja jännittäviäkin tapahtumia, sillä osa osallistujista on voinut olla mukana yhteisössä jo vuosikymmenen ajan, ellei pidempäänkin. Tapaamisissa paitisi luodaan elämänmittaisia muistoja, traumoja tai korvavaurioita, myös luo-

daan yhdessä jotain todella hienoa.

Kaistapäisempiä tilanteita lienevät olleet vaikkapa spontaanit yhteislaulut kanssaihminen ihmetykseksi esimerkiksi Kampin kauppakeskuksessa ja Turun linnassa. Niiden sivuvaikutuksena ainakin City-Centerin vartijat ovat päätyneet harkitsemaan vakavasti uranvaihtoa. Samassa hengessä simsaajat ovat kannatelleet ilmoille myös kokonaisen runokirjaprojektin ja monta erikoista taideprojektia. Eräässä näistä Blumbobin - simien pään päällä leijuvan timantin - muotoinen kakku syötiin parempiin suihin keskellä puisto. Kakku tapahtui myös toisellakin kertaa, jolloin kermää pursui pitkin Narinkkatoria.

Yhteisön kautta on myös järjestynyt useita ensimmäisiä työpaikkoja tai työkokeiluja, ja simsaajat ovat olleet vuosien ajan tuttu näky esimerkiksi DigiExpo-tapahtumassa ja pelimessuilla. Yhteisö on tarjonnut myös tukea niin omilleen muuttoon kuin moneen muuhunkin elämänvaiheeseen, sekä tietysti luonut kasapäin yhteisiä muistoja. Joskus foorumilla on kirjoitettu yhteistyönä fanifcejä yhteisön jäsenistä ja ylläpitäjistä, joskus porukka on rynnistänyt yllätyshyökkäyksenä pelivalmistajan toimistolle tai ajanut admininsa onnistuneesti läpi nokkospuskien.

Suomalaisia simssaajia on kuljetettu myös vierailuille pelivalmistajan konttoreihin niin Suomessa, Pohjoismaissa kuin maailmalla, sekä testaamaan pelejä ennakkoon että tapaamaan muita faneja ja pelikehittäjiä.

Onkin helppo todeta, että yhteisössä on voimaa, eikä siksi ole kummaa, että esimerkiksi *Radosims* on porskuttanut jo yli 12 vuoden ajan. Sivuston ylläpitäjä **Radon** mukaan tämä on paljolti ”maailman parhaiden moderaattoreiden” ansiota.

Radolan yhteisö on myös järjestänyt omia kilpailuja, osallistunut tohtorinväitöstilaisuuksiin, antanut vinoja pinoja haastatteluja, ollut mukana luomassa suomalaista pelikulttuuria ja käynyt studiovierailujen välillä aktiivisesti kirjeenvaihtoa pelivalmistajan kanssa.

Kun tämä kaikki on vielä tapahtunut turvallisessa ympäristössä, jossa häiriköintiin puututaan nopeasti ja lujasti, moni nykyinen aikuinen on huomannut kasvaneensa paitsi elämäsimulaation, myös upean yhteisön parissa.

Sometarinoita Suomesta -kirjan kirjoittaja **Jarno Alastalo** toteaaakin verkoyhteisöistä näin:

”Nettiyhteisöjen kautta on mahdollista löytää ja tavoittaa samanhenkiset ihmiset, myös silloin kun asuu kaukana. Siksi ne ovat niin hienoja.”

Sims-yhteisökin on kautta historiansa tarjonnut juttuseuraa myös sinne, missä asuu vain kolme ihmistä ja orava. Aika näyttää, miten tulevaisuus kohtelee *Sims*-yhteisöllisyyttä. Takana on kuitenkin jo kunniakas historia, joten myös tulevaisuudelta voi odottaa paljon. Vai miltä kuulostaa se, että *Simsin* komeimmista bugeista tehdään taidenäyttely? 🐿

Kirjoittaja Milsa on ollut *Sims*-aktiivi vuodesta nakki lähtien.

Linkkejä

Radosims

radosims.com

Radola-keskustelufoorumi

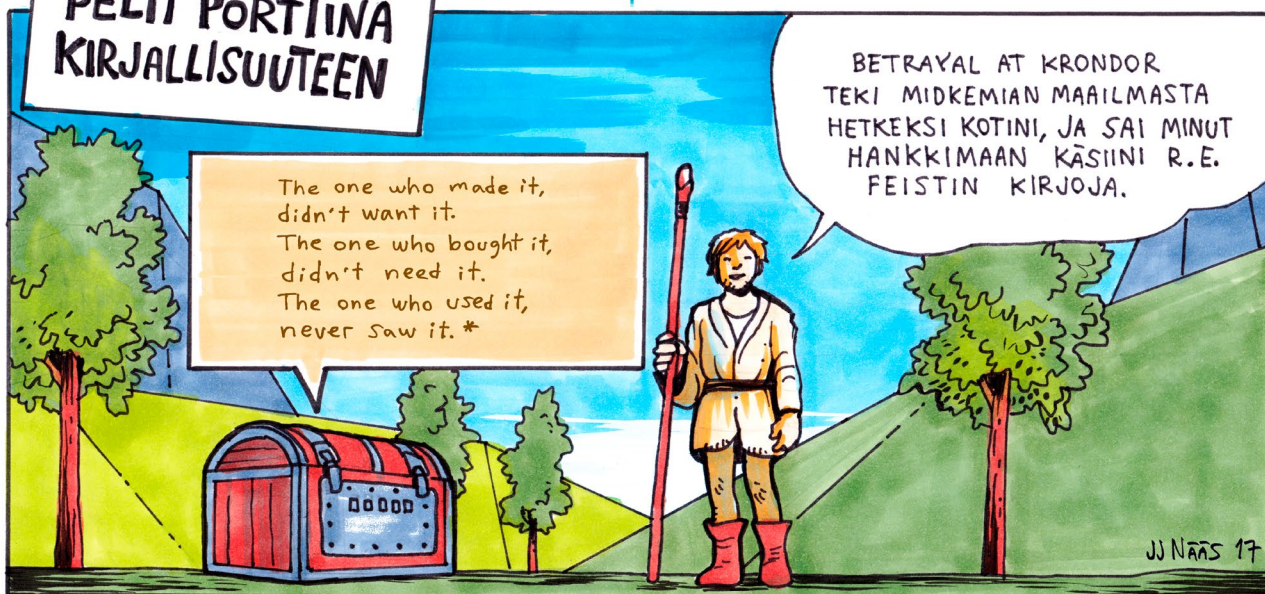
www.radosims.com/radola/

Sometarinoita Suomesta

www.adlibris.com/fi/kirja/sometarinoita-suomesta-9789523040557



PELIT PORTTINA KIRJALLISUUTEEN

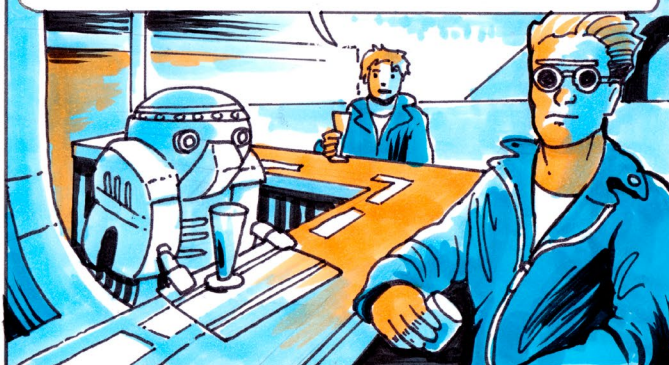


LÄHIKIRJASTOSTA LÖYTYI KOKO
RIFTWAR -SAAGA, PISTEET
SIITÄ MUSTASAAREN KUNNAN-
KIRJASTOLLE.

FEISTIN MAGICIAN OLI
ENSIMMÄINEN ENGLANNIKSI
LUKEMANI KIRJA.



FREDERICK POHL'S GATEWAY -PELISSÄ
EN PÄÄSSYT ETENEMÄÄN KOVINKAAN
PITKÄLLE. KIRJAN LUIN SILTI, JA SEN
TAPAHTUMAT OVAT EDELLEEN SELKEINÄ
MIELESSÄ.



H.P. LOVECRAFT OLI MINULLE TUNTEMATON
KIRJAILIJA ENNEN ALONE IN THE DARK
-PELEJÄ. THE HAUNTER OF THE DARK
-NOVELLI VEI YÖUNENI, MIKÄÄN
KAUHUPELI EI OLE SIIHEN VIELÄ
KYENNYT.



DRAGONLANCET
OLISIN VARMASTI
LUKENUK ILMAN
SSI:N PELEJÄKIN.



DYYNIN LUIN ENSIM-
MÄISEN DUNEN
PELIARVOSTELUN
INSPIROIMANA,
ITSE PELIÄ EN OLE
KOSKAAN PELANNUT.



*coffin

Wolfenstein 3D oli rankka, raaka, pyöräyttävän nopea ja niin vauhdikas, että jotkut voivat pahoin. Se näytti uskomattomalta ja kuulosti hämmästyttävältä. Se aseiden pauke, vihollisten huudot, kuoleman örinät. Natsit, natsikoirat ja Hitler. Niistä on rakas klassikko tehty.

Teksti: Jukka O. Kauppinen
Kuvat: Mobygames, id Software, Bethesda Softworks



WOLFENSTEININ PERINTÖ

Guten Tag, mein KyberFührer

Neljännesvuosisata sitten videopelit muuttuivat. Mikään ei ollut enää entisensä, kun William B. J. Blazkowicz antoi natseille turpaan.

5.1992 Wolfenstein 3D -räiskintäpeli laskeutui pelimaailmaan. Maailma oli tuolloin kovin erilainen. Amiga oli pelikukkulan kuningas. Kuningatar Elisabeth II juhli 40-vuotista hallituskauttaan. Konsolimaailmaa hallitsivat Segan MegaDrive ja Nintendon SNES, mutta moni nautiskeli yhä myös Commodore 64:n, NESin ja Master Systemin peleistä. Bill Clinton valittiin USA:n presidentiksi. Jossain tikitti yksinäinen ColecoVision.

Mutta PC? Jaa se tekstinkäsittelylaitte? Se, jolle ei ollut pelejä ja jos olikin, niin ne olivat rumia ja kurjia ja kökköjä. Amiga Gaming Master Race hallitsi, sillä jos halusit näyttäviä pelejä upealla grafiikalla, messevillä äänillä ja mahdollalla ruudunpäivityksellä, niin muita järkeviä vaihtoehtoja ei ollut.

Amigistit, konsolistit ja jopa kuusnelospelaajat katselivat PC-pelaajaparkoja säälien, kun nämä yrittivät väkisin pelata nykiviä tasoloikkiaan. Niiden äänetkin olivat jos ei piipitystä, niin särinää täynnä. Ei ollut elämää AmigaOS:n tuolla puolen.

Mutta Yhdysvalloissa oli pieni, nuori peliyhtiö, jonka riveissä oli muutama

nuori, aina vain korkeammalle tähtäävä pelintekijä. He olivat hioneet taitojaan toisten palkkalistoilla, oppineet tekemään töitä yhdessä ja löytäneet toisistaan niin tukea, inspiraatiota kuin raivokasta tekemisen iloa. Poppoon jäsenet olivat kovia pelaajia ja osaajia ja halusivat kehittää pelintekoa eteenpäin aivan omanlaisiin suuntiinsa.

Ja kyllähän se tältä sakilta sujui. Firman miehitys vaihteli jonkin verran, mutta legendaarisessa ydintiimissä **Tom Hall** vastasi suunnittelusta, siinä missä pitkäletinen **John Romero** päti sekä ohjelmoijana, ideoijana että

pelisuunnittelijana siirtyen peliltä enemmän idea- ja suunnittelusuuntaan. **Adrian Carmack** taikoi grafiikkaa, mutta porukan valttikortiksi muotoutui se carmackimpi John, sosiaalisista käytöstavoista vähät välittävä koodaajanero. **John Carmack** osasi katsoa asioita eri suunnasta kuin muut, löytää kikkoja teknologian hyödyntämiseen ja kehittää uutta, sen sijaan että vain kopioisi muiden tekemisiä.

Tämä tiivis ryhmä muodosti pelijengin, joka puhui samaa kieltä ja hengitti samaa ilmaa. Usein asuikin yhdessä. Elämä oli peliä 24/7.



Wolfenstein leikkeli – tai pikemmin pilkkasi – natsismilla niin avoimesti, että peli kiellettiin kertalokasta niin Saksan kaupoista kuin internetistä.

Kohti kolmea deetä

Tiimin tekniset kokeilut siirtyivät jo varhaisessa vaiheessa sivusta ja päältä kuvatuista peleistä ja yksinkertaisemmista toimintapeleistä kohti uusia suuntauksia. Voisiko näillä koneenrämillä – tai siis sen aikaisilla huippukoneilla – luoda myös ihan oikeaa 3D-grafiikkaa?

3D:tä oli toki tehty ennenkin. Esimerkiksi vuoden 1985 Elite pyöri ihan siistillä 3D:llä, tosin viivawireframena, ja joissakin versioissa objekteista näkyi läpi. Bard's Tale- ja Ultima-roolipelit lanseerasivat tietynlaisen naamaperspektiivin, tosin vain pienellä nurkaan sijoitetulla ruudulla.

John Carmackin koodarintaidot olivat näihin aikoihin hurjassa terässä. Id Softwaren pojat loivat pelejä eläimelliseen tahtiin, sillä heidän piti tehdä Softdisk-julkaisijalle uusi peli joka toinen kuukausi. Samaan aikaan he työskentelivät myös Apogee-kustantajan kanssa. Porukka teki peräkanaa useita pelisarjoja, joten rutiinia oli pakko löytä. Jossain välissä Johnit ohjelmoivat ensimmäisen 3D-pelinsä, Hovertank 3D:n. Urakka oli valtava, sillä Johneilta meni kokonaiset kuusi viikkoa pelimoottorin tekemiseen.

Heti perään syntyi Catacomb 3-D, inspiraationaan Romeron kaverin työstämä seikkailu Ultima Underworld. Catacombissa Hoverin yksiväriset seinät teksturoitiin Ultiman haastamana värikkäästi ja näyttävästi, samalla kun sankari juoksenteli sokkeiloissa FPS-hengessä ammuskellen vastaan tulevia örkkejä tulipalloilla.

Ainekset videopelien vallankumoukseen olivat valmiit. Enää tarvittiin vain lopullinen sysäys.

Tapa natsi, tapa kaksi, tee maailma paremmaksi

Apogeen kustantaja Scott Miller antoi alkutönäyksen, kun hän näki Catacombsin.

”Meidän täytyy tehdä jotain tällaista”, meuhkasi sharewarella mahtitajoja takova Miller. Sharewaressa pelaajille annettiin ilmaiseksi osa peliä, mutta loput sai ostamalla. Taktiikka toimi hillittömän hyvin ja raha virtasi.

Sen myötä Tom Hall mietti millaisen toimintapelin Carmackin pelimoottorilla voisi tehdä. Jos olisi hirviötä, mutantteja, 3D-demoneja ja vihreitä limatuksia. Romero ei lämmennyt, vaan vaati että ”nyt tehdään jotain siistiä. Eikö olisi helkkarin upeaa, jos vaikka tekisimme Castle Wolfensteinin uuden version, mutta 3D:nä?”

Siitä se lähti.

Wolf oli alusta alkaen 3D-peli, mutta räiskinnän piti olla vain sivuseikka. Alun perin tiimi teki stealth-taktiikkapeliä, vaikka jossain välissä kulma oli myös avaruudellisemmassa scifiräiskinnässä Aliens-elokuvan hengessä. Lopulta kulma tarkentui natseihin ja linnoihin. Samalla Carmackin viilattu pelimoottori osoittautui häikäisevän nopeaksi ja näyttäväksi. Pian kaikki ylimääräinen pudotettiin pois ja jäljelle jätettiin vain aiempien designien raamit – peli, jossa oli käytännössä yksi ainoa timanginen idea. Raivokas, aggressiivinen ja nopea räiskintäpeli, jollaista ei oltu ennen nähty.

Kuningasidean muovautumisen jälkeen peli syntyi kolmessa kuukaudessa. Molemmat Johnit sukelsivat koodin uumeniin. Romero ragetti raivoräiskinnän perään ja Tom Hall rakensi kenttiä. Ideat lentelivät, usein täynnä verta ja suolenpätkiä. Läheskään kaikkea ei peliin laitettu, eikä sankari voinut lopullisessa pelissä parantaa itseään syömällä tapettujen vihollisten sisälmyksiä.

Mutta brutaali se oli – täynnä Romeron visiota, Carmackin koodia, Adrianin natseja ja Hallin kenttiä.

4.5.1992 modeemi lauloi Software Creations BBS:lle ja aloitti siirron.

Pelaajien muistoja

Pyysimme pelaajia kertomaan muistojaan Wolfenstein 3D:stä niin Facebookissa kuin keskustelupalstallamme.



Hybabo! *rätätätätätä*
Mein leben!
Jaakko Vähälä



Atari ST:llä on tullut pelailtua. Oli hämmästyttävää miten peli on saatu pyörimään 8 megahertsin raudalla.
Marko Latvanen

Wolfensteinin valmis shareware-painos aloitti maailmanympärimatkansa, ja Apogee valmistautui ottamaan vastaan tilauksia täysversiosta.

”Jos pelaajat tykkäävät tästä, niin Wolfenstein saattaisi myydä jopa tuplasti Commander Keeniin verrattuna”, haaveili Tom Hall. Keenit olivat pitäneet pitkään shareware-listojen ykkössijaa, ja taanneet pojille turvaton elannon noin 30 000 dollarin kuukausitilityksellään (nykyrahassa noin 54 000 USD, 51 000 €).

Wolfenstein 3D:n ensimmäisen kuukauden tilitys oli 100 000 dollaria. Sen tekeminen oli maksanut 25 000.

Wolfenstein oli välittömästi omaa luokkaansa

Ripaus uutta, ripaus vanhaa ja järjettömästi vauhtia

Wolfenstein 3D oli peli, joka jäi mieleen. Se ei kuitenkaan tullut aivan niin yllättäen ja tyhjältä, kuin moni sen ajan pelaaja koki. Hovertank ja Catacombs kun viitoittivat selvän kehityspolun kohti visuaalisesti näyttä-

Se oikea Wolfenstein

Wolfenstein 3D on ensimmäinen id Softwaren eepisistä 3D-räiskinnöistä mutta ei suinkaan ensimmäinen Wolfenstein. Se kunnia menee näet Muse Softwaren tekemille Castle Wolfenstein -hiiviskelypeleille, jotka saattoivat jopa luoda stealth-hiippailugenren. Niissäkin kuljeltiin natsien täyttämässä linnakkeissa, mutta räväkän räiskeen sijaan pelaaja yritti pysyä piilossa ja nitistää natsit huomaamatta.

Castle Wolfenstein ja jatko-osa Beyond Castle Wolfenstein olivat varsin kovia hittejä Applella ja sittemmin muun muassa C-64:llä. Ne innoittivat myös id:n poikia, jotka halusivat tehdä oman tulkintansa natsihiippailuräiskinnästä. Pelille visioitiin kaikenlaisia nimiä, mutta sitten he huomasivat Wolfenstein-nimen tekijänoikeuksien rauenneen Musen konkurssissa. Niinpä kaverit vain kaappasivat nimen itselleen, ja loppu onkin historiaa.

TRIVIA

Mikä näistä oli Wolfenstein 3D:n nimi jossain vaiheessa kehitystä?

Adolph's Bane

Castle Hasselhoff

The Fourth Reich

It's Green and Pissed

Luger's Run

Vastaus jutun lopussa.

vämpiä, vauhdikkaampia räiskintöjä, mutta kumpikaan niistä ei ollut varsinaisesti läpimurto- tai hittipeli, ei etenäkään suomalaisilla raukoilla rajoilla.

Wolf oli myös se peli, joka lopullisesti räjäytti sharewaren suuren yleisön tietoisuuteen. Se ei suinkaan keksinyt ”ensin tötsyt, sitten rahat” -taktiikkaa, mutta Wolfensteinin upeus sai suomalaisetkin selvittämään, miten sinne Amerikkaan oikein laitetaan rahaa. Onneksi boksipainokset, sharewarekauppiat ja tietysti rehellinen ware helpottivat himoräiskijöiden elämää, kuten myös puolen vuoden sisään julkaistu lisä- ja jatko-osa.

Vaan mikä oli Wolfensteinin menestyksen salaisuus? Ei mikään yksittäinen juttu, vaan monien asioiden onnistuminen oikeaan aikaan tavalla, joka oli jotain aivan uutta, odottamatonta ja uskomatonta.

Esimerkiksi väkivalta ja natsimurhajuhat osuivat kipeään ajankohtaan, kun videopelien väkivaltaa ja haitallisuutta lapsille puntaroitiin Yhdysvaltain senaatissa saakka. Wolf sai polemiikkista ilmaista mainosta, eivätkä id Softwaren jampat paljoa muiden mielipiteistä pii-



tanneet. He tekivät omaa juttuaan.

Ja se olikin melkoinen juttu se. Saksa-laissotilaat huusivat saksaksi toisilleen ja kuollessaan äitiä. Hitler hyvästeli Evan. Alkuruudussa soi Saksassa kielletty sotilasmarssi. Veri pulppusi ja vankisellit olivat täynnä vankien luurankoja. Sotakoirat kuolivat vinkaisten ja Death Cam näytti episodien loppubossien kuoleman hidastettuna uusintana, kameran kiertäessä ympärillä kuin snuff-leffassa ikään.

Siinä samalla lukemattomat pelaajat oppivat juoksemaan pitkin seinää välilyöntiä takoen. Jostain se salakäytävä ja piilohuone aina löytyi. Ai, kentällä on vielä kolme löytämätöntä salaisuutta – ja alkuun, kyllä ne vielä paljastuvat.

Wolfensteinista kasvoi ilmiö. Peliä kehuttiin jopa virtuaalitelokkuudeksi ja paikaksi, joka on niin realistinen, että sinne saattaa unohtua. Se oli niin ennennäkemättömän nopea, että jotkut voivat pahoin ja oksensivat pelat-



Tulee mieleen kyllä muistoja. Koulussa muokattiin peliä apuohjelmalla, jolla sai muutettua grafiikkaa. Ase, jolla natsveja ammuttiin, muutettiin valkoista ainetta ampuvaksi k*rväksi. Kuolleiden natsien verilammitkot muutettiin valkoisiksi ja jokaiselle kuolleelle natsille jäi k*rpä pystyyn. Oli kyllä paljon hauskeampi sen jälkeen...

Marko Toivanen



Koulussa pelattiin 75% ruudulla Mikromikko 3TT:llä. Ilmeisesti koneessa oli 286/12mhz ja VGA. Ei se silläkään sulavasti pyörinyt ja äänet tulivat PC-kaiuttimesta.

Teemu Korvenpää



Wolfensteineja neljällä vuosikymmenellä

Castle Wolfenstein (1981)

Beyond Castle Wolfenstein (1984)

Wolfenstein 3D (1992)

Spear of Destiny (1992)

Return to Castle Wolfenstein (2001)

Wolfenstein: Enemy Territory (2003)

Wolfenstein RPG (2008)

Wolfenstein (2009)

Wolfenstein: The New Order (2014)

Wolfenstein: The Old Blood (2015)

Wolfenstein: New Order reboottasi Wolfenstein-sarjan nykyaikaan tyyllillä, asenteella ja isoilla aseilla. Samalla se oli myös upea ja ajatuksia herättävä tarina, mitä ei alkuperäispelistä voi väittää.

tuaan liikaa. Se oli – väitetään mitä tahansa – ensimmäinen todellinen, moderni FPS-räiskintäpeli.

Se asetti myös genren standardit, joista monia noudatetaan yhä tänään. Kuinka pyssy näkyy ruudulla. Kenttien rakenne, sokkeloisuus ja salahuoneiden etsiminen sataprosenttisen läpäisyn toivossa. Siirtyminen pyssyleikeistä lähitaisteluun ja ruumiinpalojen lentely. Räiskintähistorian klassiset perusaseet pistoolista konekivääriin ja raketinheittimeen. Yksinäisen badass-sankarin nostaminen pääosaan. Kaiken kukkuraksi se oli tietenkin ensimmäinen toisen maailmansodan FPS. Syökää siitä, CoDit ja Bäfät!

Eikä kestänyt kauaa, kun Wolfensteinia modattiin tosissaan. Sitä ei ollut koskaan tarkoitettu modat-

tavaksi, mutta ei se säätäjiä estänyt, vain hidasti. Wolfensteinin fanikentät, total conversionit, pelaamista värittävät tai helpottavat modit ja muut viritykset herättivät myös id Softwarin huomion, pääosin positiivisella tapaa. Firmassa tajuttiin, että jos fanit kerran haluavat näprätä, niin annetaan siihen mahdollisuus. Niinpä myöhemmät Doom ja Quake avattiin alusta lähtien pelaajien räplättäväksi. Modatkaa, modatkaa minkä vain jaksatte, id huusi. Ja pelaajat modasivat.

Id Softwarin pojat olivat nyt superhäntiä. He jopa antoivat itselleen palkankorotuksen: 9000:sta taalasta per nokka 45 000:een vuodessa. Elämä hymyili. Videopelit eivät olleet enää entisensä. 🍷



Moniko teistä löysi Wolfensteinin Pac-Man-salakentän?



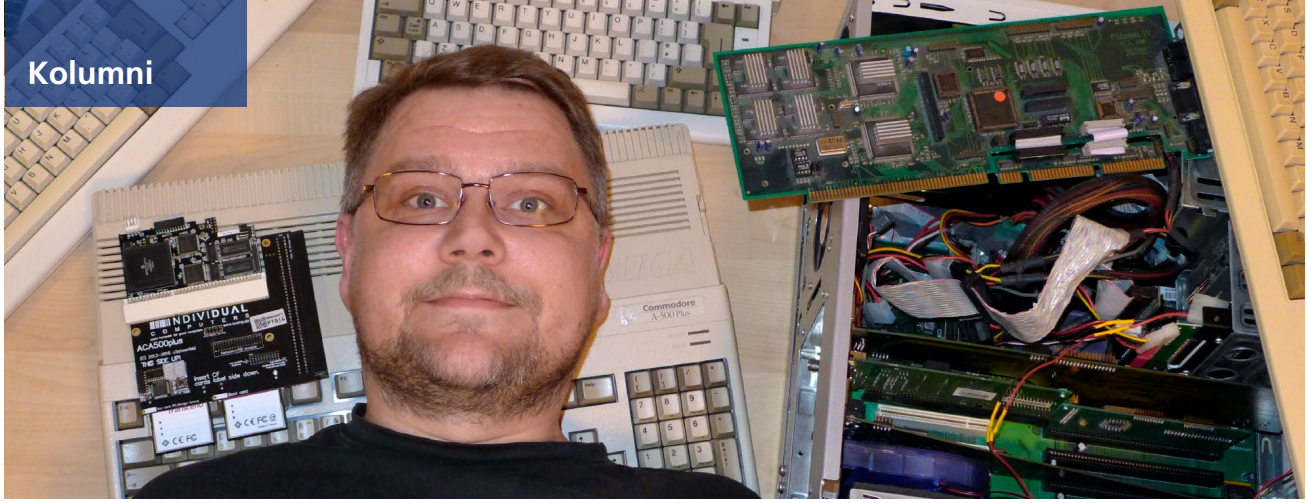
Jännä peli, jota hakattiin tuolin reunalla istuen adrenaliinin pauhatessa. Ei sellaista oltu ennen nähty. Wolf ja jatko-osa Spear of Destiny olivat vaikeita, en tainnut teininä päästä kaikkia episodeja läpi normivaikeustasolla. Doomintultua nämä jäivät, mutta tuli niiden pariin vielä myöhemmin palattua. Wolfensteinissa oli vielä oikeasti vaikeutta. Doomissa ensimmäisestä singon osumastakin selviää vielä hengissä, mutta kovalla vaikeustasolla Wolfissa lähti henki kahdesta lähietäisyydeltä ammutusta kudista – ja sellaisia varsinkin valkopukuiset upseerit olivat nopeita jakelemaan.

Tapani Harkila



Takaapäin yllättävät viholliset olivat hauskoja, samaten Death-Cam. Demoista tuli ihmeteltyä, että miten toi ukko pystyy kääntymään noin nopeasti ja tähtäämään tarkasti. Ne oli pelattu hiirellä, mikä oli vielä tuossa vaiheessa aikajanaa täysin dokumentoimaton ominaisuus. Sitä ei edes löytynyt kasuaalisista options-valikoista. Suuret massat paukuttelivat näppiksellä vielä Doomiakin, missä setupissa kikkailemalla tai erillisillä dos-ohjelmilla sai hiiritähtäyksen päälle. Ei kasuaali-ihmisten toimintaa sekään. Olikohan Quake 1 suunniteltu ja tarkoitettu kentiltään ja aseiltaan vain näppikseltä pelattavaksi? Hiiren kun voi enableoida vain autoexec.cfg:stä lisäämällä skriptin loppuun +mlook-komennon. Ei kukaan normaali-ihminen moista osannut tehdä. Piti mennä vielä Doominkin jälkeen Quakeen asti ennen kuin hiirellä pelaaminen saatiin kasuaalipelaajien ulottuville, silloinkin harrastajien toimesta. Vasta Quake II:ssa oli options-valikossa hiiritähtäyksen käyttäminen, mikä on harvinaisen oimtuinen sivujuoni id:n pelien saagassa, etenkin kun muistaa millainen käsite hiiritähtäys on FPS-peleissä.

Jani Ikävalko



Ihana, kamala *Amiga*

Mikko Heinonen

Siinä se taas kului ilta, kiroillessa ja tuskanhikeä pyyhkiessä. Posti toi Puolasta paketin, joka sisälsi päivitettäväksi lähettämäni Amiga 1200:n turbokortin. Sen reaaliarvo vapailla markkinoilla olisi todennäköisesti suurempi kuin nykyaikaisen Windows-tietokoneen, jolla kirjoitan tätä juttua. Omistin jo ennestään kaksi 68060-pohjaista turboa, mutta hyvä se kolmaskin on olla. Jos vaikka kasvatan lisää päitä.

Toinen paketti sisälsi kiintolevyohjaimen, joka pyrkii ratkaisemaan toista A1200:n perusongelmaa: tukkoista levykäsitteilyä. Ohjain oli tarkoitus asentaa samaan koneeseen mainitun turbokortin ja 3D-kiihdytetyn näytönohjaimen kanssa, mutta eiväthän ne tietenkään mahdu paikoilleen yhtä aikaa. Amiga 1200:n emolevy ja pöytäkotelo ovat liian pieniä, eikä laitetta ole alkujaankaan tarkoitettu aivan näin laajennettavaksi.

Ratkaisu oli hankkia ulkomailta korotuspalanen, joka nostaa levyohjaimen vinoon ja mahdollistaa näytönohjaimen asentamisen sen alle. Muuten hyvä, mutta korokkeen laadunvarmistus oli pettänyt, eikä signaali edes kulkenut kaikkien sen nastojen läpi. Ja vaikka olisikin kulkenut, sen kiinnitysmekanismi tuntui perustuvan etupäässä lujaan uskoon. Sain koneen käynnistymään, sitten siirsin emolevyä hieman ja se mykistyi uudelleen. Lopulta päädyin juottamaan koneen Kickstart-ROMit kiinteästi korokkeeseen kiinni ja rakentamaan uskomattoman nippusidevirityksen pitämään koko tötteröä paikoillaan. Tämän jälkeen osa-

kaasa siirtyi sivupöydälle odottamaan jälleen sitä hetkeä, kun koko projekti ei saa lähinnä näkemään punaista.

Oi, kuinka häilyvä

Tämä A1200-projekti ei kuitenkaan ole edes likimainkaan epätoivoisin Amiga-rakenteluni. Kun vihdoin pääsin elämässäni siihen tilanteeseen, että sain säännöllistä palkkaa, eräs tärkeimmistä projekteista oli hankkia kaikki se Amiga-rauta, jota en nuorempana rahapulassa voinut ostaa. Tästä syntyi Paksu Bertta. Kasasin Amiga 4000:n emolevyn ympärille – kustannuksista välittämättä – kaikki lisälaitteet, joista olin lukenut aikoinaan lehdistä.

Kokosin Paksun Bertan ensi kertaa yli 10 vuotta sitten, eikä se ole tähän mennessä toiminut hetkeäkään niin kuin alkujaan ajattelin. Lisälaitteiden välillä on ollut konflikteja, jotka lähentelevät mustaa magiaa. Näytönohjain on rikki, vaikka se kävi Ranskassa asti mukamas korjattavana. Innostuksissani olin jossain vaiheessa onnistunut hankkimaan PowerPC-suorittimen ytimen ja 68060 muuten vain irtosi juotoksistaan, joten tämänkin koneen turbokortti piti lähettää Puolaan kalliiseen remonttiin. Rahaa oli jo sitä ennen palanut nelinumeroinen summa.

Voisi kuvitella, että olisin jo aikaa sitten listannut molempien koneiden palaset myyntiin verkkohuutokauppaan. Koska olin ajoissa liikkeellä niitä ostessani, saisin helposti rahani takaisin korjauskustannustenkin jälkeen. Mutta rahalla saa lopulta keskinkertaisia asioita, ja jos luopuisin näistä hienoista laajennuksista, harmittelisin sitä ta-

kuuvarmasti. Jo pelkkä tieto siitä, että minulla on ne, saa hymyilemään.

Rajoja ja rakkautta

Facebookin Amiga-ryhmässä muuan ulkomainen harrastaja kyseli taannoin, miksi ihmeessä porukka täällä kasaa kauhean hintaisia Classic Amiga -settejä, kun emulaattorilla varustettu nykyaikainen tietokone tekee kaiken yhtä nopeasti, ilmaiseksi ja paljon vähemmällä vaivalla. Kysely sai aikaan vahvan reaktion, eikä ihme. Sama olisi ollut kysyä, miksi ihmeessä porukka on kiinnostunut vanhojen autojen ruuvaamisesta, kun uusiakin myydään.

Tiedän, että Amiga-harrastuksessani ei ole taloudellista järkeä. Vaikka periaatteessa voisin päästä voitolle luopumalla nyt kaikesta omistamastani, tiedän samalla, etten tule niin koskaan tekemään. Vähitellen kaltaiseni innokkaimmat amigistit häviävät ja palikoiden hinta kääntyy laskuun. Katson kuitenkin olevani voitolla – sillä silloin, kun koneprojektit eivät raivostuta, ne antavat mitä korkeinta tyydytystä.

Sählätessäni juotuskolvin kanssa Amigan emolevyllä koen toimivani taitojeni rajoilla. On olemassa mahdollisuus, että saan aikaiseksi hintavan verkonpainon tai seinäkoristeen. Mutta on myös mahdollista, että tämän joskus vuosiakin kestävä virittelyn päätteeksi minulla lopulta on juuri sellainen laite, josta sain aikoinaan lukea unelmatestejä harrastelehdistä. Voin ajaa niitä muutamaa kymmentä RTG-grafiikkapeliä 1990-luvun puolivälistä ja nauttia skenedemot täydellä nopeudella muualtakin kuin YouTubeista. Ja mikä tärkeintä: käpistelystä syntyy vastapainoa normaalille aivotyölle ja kirjoittamiselle. Aina silloin kun en vihaa Amigoitani, rakastan niitä. 🐱

Lisänostetta Haukan liitton

CT60e

Atari Falconin 68060-pohjainen kiihdytin

Atari Falcon jäi legendaarisen kotimikrovalmistajan viimeiseksi tuotteeksi, ja innokkaiden harrastajien ansiosta se elää tänäkin päivänä. Alkuperäinen 68030-prosessori ei enää uusimpiin ohjelmiin riitä, mutta onneksi saatavana on lisävauhtia.

Teksti: Joni Valtanen

Kuvat: Joni Valtanen, Mika Yrjölä



CT60e on Atari Falconin vastine Amigalta tutuille 68060-kiihdytinkorteille. Siinä on siis mahdollista korvata Atari Falconin Motorola 68030 -prosessori tehokkaammalla 68060-prosessorilla ja samalla lisätä koneen muistia jopa 512 megatavulla. Muisti tulee näkyviin Atari TT030 -koneista tutuksi tulleen ALT-RAMina, johon ST-yhteensopiva osa piirisarjasta ei pääse käsiksi. Kortista on ollut aikoinaan saatavilla CT60- ja CT63-versiot. E-kirjain tulee evoluutio-sanasta, joka melko hyvin kuvastaa kortin yritystä olla edeltäjiään parempi. Atari Falconin alkuperäisen 68030-prosessorin kellotaajuus on 16 MHz. CT60e-kortissa olevan 68060-prosessorin kellotaajuutta voidaan säätää 66:stä 110 megahertsiin – tosin kaikki prosessoriversiot eivät aivan huippulukemiin kykene. Suorituskykyisyys on merkittävä: jo pelkästään 68060-prosessorin vaihtaminen 68030-prosessorin tilalle tekisi koneesta jonkin verran nopeamman, mutta lisäksi koneen kellotaajuus yli kuusinkertaistuu.

CT60e:n synty

Nykyisten CT60e-korttien syntytarina on monivaiheinen. Ennen vuotta 2014 alkuperäisten CT60-korttien suunnittelija **Rodolphe Czuba** oli harkinnut julkaisevansa CTX60-nimellä

kulkevan kortin. Kortissa olisi kaiketi yhdistynyt CT60 ja CTPCI-niminen PCI-väylälajennus. Vuonna 2014 Czuba kuitenkin ilmoitti, että useampikin ihminen oli häneltä kysellyt alkuperäisten CT60-korttien perään ja nyt hän lähetti ihmisille tiedusteluja siitä, olisivatko he kiinnostuneita kyseisistä korteista. Myöhemmin samana vuonna alkoi kiertää huhuja CT60e-korteista. Joulukuussa Czuba varmisti huhut tosiksi ja samalla kertoi, että hän itse ei kortteja valmistaisi.

Willy-nimimerkillä toimiva mies esittäytyi 2015 vuonna CT60-korttien jatkokehittäjäksi ja saikin jonkin ajan kuluttua tehtyä prototyypin CT60e-kortista. Prototyypikortissa esiintyi ongelmia ja Willy katosi maan alle useammaksi kuukaudeksi. Maaliskuussa 2016 Willy ilmoitti, että oli saanut kortin vihdoinkin toimimaan. Tämän jälkeen hän katosi jälleen foorumeilta. Puolen vuoden kuluttua tuli kuin tyhjästä ilmoitus, että kortit ovat nyt tuotannossa ja ensimmäinen tuotantoerä olisi 100 kappaletta. Vihdoinkin maaliskuussa 2017 kortit tulivat myyntiin ja niitä lähetettiin ennakkotilajille, joita oli ollut jopa noin 50 kappaletta.

Käyttäjien toiveiden ja käytännön pulmien vuoksi CT60e-korttia on paranneltu alkuperäisestä CT60:stä. Piirilevyllä on lisätty kerroksia maadoitusvedoille, jotta kortista tulisi

vähemmän altis häiriöille. Myös mekaanista toimivuutta on paranneltu. CT60e-kortti kiinnittyy Falconin väylän (Falcon Bus) lisäksi virtaliittimeen ja kortissa on reiät yläpuolisten kuoren kiinnitystappien läpivientiin. Aiemmat versiot kortista olivat kiinni ainoastaan Falconin väylässä ja siirrettäessä ne saattoivat lähteä helposti paikoiltaan, minkä vuoksi useampi ihminen oli ilmeisesti onnistunut käräyttämään Falconinsa tai CT60:nsä.

CT60e:ssä on integroitu CTCM eli säädettävä prosessorin kellotaajuusyksikkö ja edeltäjiensä tapaan 060-väylä. 060-väylään on mahdollista liittää erilaisia lisälaitteita, kuten näytönohjain-, verkko- ja USB-kortti sekä PCI-väyläkortti. CT60e ei tosin ole mekaanisesti yhteensopiva CTPCI-nimisen PCI-väylälajennuksen kanssa. Kortin asetukset tallentava flash-piiri on myös vaihdettu ja tämän vuoksi vanhat päivitysohjelmat eivät toimi.

Testikoneen asennus

CT60e-kortti on edeltäjiään hieman kookkaampi, jotta 060-väylä saadaan sijoitettua hieman taaemmaksi. Uusi sijoittelu auttaa esimerkiksi SuperVidel-näytönohjaimen liittämisessä ilman lisäosia Atari Falconin koteloon. Aiemmin CT60-kortin 060-väylä meni osaksi näppäimistön alle ja ainoastaan yksi kortti mahtui 060-väylään. Täl-

löinkin oli hyvä irrottaa ruuveja näppäimistön pohjasta, jotta ei syntyisi oikosulkua. Uuden suunnittelun ansiosta 060-väylään on kaiketi mahdollista liittää jopa kaksi korttia päällekkäin Falconin koteloon.

Ennen CT60e-kortin asennusta Atari Falconista pitää poistaa virtalähde ja metallisuojaus emolevyn päältä. Virtalähde korvataan ATX-virtalähteellä. Pienikokoisessa 120 watin Pico-ATX-virtalähteessä on enemmän kuin tarpeeksi tehoa Falconin, CT60e-kortin ja siihen liitettävien lisäkorttien tarpeisiin.

Asennukseen vaaditaan myös on-off-kytkin ja yksi lukkiutumaton painonappi. Korteissa olevasta liittimestä löytyy ATX:n virtakytkin, reset-näppäin sekä kytkin, joka kytkee koko CT60e:n 030- tai 060-tilaan. Reset-näppäin on näppärä, mikäli Falcon halutaan tornittaa. Jos CT60e kytketään 030-tilaan, Falcon käyttää omaa 68030-prosessoria ja ainoastaan omaa muistiaan. CT60e ja siihen liitetyt lisälaitteet ovat poissa käytöstä. 030-tilaa tarvitaan esimerkiksi päivittäessä CT60e-kortin flash-muisti. Tämä on myös hyvä tila ohjelmille ja peleille, jotka eivät jostain syystä CT60e-kiihdyttimen kanssa toimi.

Yhteensopivan ohjelmiston asennus

Ohjelmiston asennus on melko helppoa: aiemman FreeMiNT-käyttöjärjestelmän asennuksen 030-versio pitää vain vaihtaa 060-versioksi. Tämä tarkoittaa sitä, että mint030.prg:n sijaan käyttöön vaihdetaan mint060.prg. Mikäli FreeMiNTin versio on 1.17 tai sitä vanhempi, pitää myös XaAES:n FreeMiNT-kernelmoduuli vaihtaa 060-versioon. Versio 1.18 ja sitä uudemmat lataavat oikean moduulin riippuen mint.prg:n versiosta. Työpöytäohjelmistona voi käyttää esimerkiksi Teradeskiä, josta on ainoastaan yksi versio kaikille 68k-prosessoreille. Ensimmäiset testit CT60e-kortilla FreeMiNTin kanssa eivät luvanneet hyvää. Kaksi asiaa häiritsi 68060-prosessorin ollessa käytössä. Ensinnäkin ruudulle ilmestyi grafiikkahäiriöitä ja toisekseen Teradesk saattoi kaatua esimerkiksi kopioidessa tiedostoja. Pelkän TOS:n kanssa ilmeni ainoastaan grafiikkahäiriöitä. Teradeskin kaatuilun syyksi ilmeni huono kontakti muistin ja kortin välillä. Liitin oli melko tiukka. Kokeilemalla eri muistipiirejä

GEMBench **File** Test Extra Help

GEMBENCH 6			
Test	Time	Ratio	Statistics
GEM Dialog Box:	0.985	1334%	Display: 3446%
VDI Text:	0.150	9750%	CPU: 2848%
VDI Text Effects:	0.325	6632%	Average: 3318%
VDI Small Text:	0.320	7065%	
VDI Graphics:	0.575	4848%	
GEM Window:	2.095	477%	
Integer Division:	0.680	4627%	
Float Math:	9.220	0%	
RAM Access:	11.225	323%	
ROM Access:	2.075	3594%	
Blitting:	0.500	722%	
VDI Scroll:	1.575	608%	
Justified Text:	3.635	319%	
VDI Enquire:	0.985	2103%	
ALT RAM (FAST RAM)	1.795	4052%	

Reference	System
TOS 2.06 AES 3.20	TOS 4.04 AES 3.40
BLITTER Enabled	BLITTER NP
RES 640*200*4	RES 640*200*4
GEMDOS v0.20	GEMDOS v0.30
STE 68000	Falcon 68060
FPU Not present	Internal FPU
NUDI not present	NUDI 5.03 present
CACHE D=OFF I=OFF	CACHE D=ON I=ON
Ran from ST-RAM	Ran from ST-RAM

Gembench-testi: Atari STE (68000, 8 MHz) verrattuna CT60e:hen (68060, 100 MHz).

GEMBench **File** Test Extra Help

GEMBENCH 6			
Test	Time	Ratio	Statistics
GEM Dialog Box:	0.985	950%	Display: 2056%
VDI Text:	0.150	6467%	CPU: 846%
VDI Text Effects:	0.325	3792%	Average: 1710%
VDI Small Text:	0.320	4720%	
VDI Graphics:	0.575	1863%	
GEM Window:	2.095	390%	
Integer Division:	0.680	834%	
Float Math:	9.220	79%	
RAM Access:	11.225	167%	
ROM Access:	2.075	2307%	
Blitting:	0.500	433%	
VDI Scroll:	1.575	395%	
Justified Text:	3.635	212%	
VDI Enquire:	0.985	1342%	
ALT RAM (FAST RAM)	1.795	0%	

Reference	System
TOS 4.04 AES 3.40	TOS 4.04 AES 3.40
BLITTER Enabled	BLITTER NP
RES 640*200*4	RES 640*200*4
GEMDOS v0.30	GEMDOS v0.30
Falcon 68030	Falcon 68060
LineF FPU inst.	Internal FPU
NUDI not present	NUDI 5.03 present
CACHE D=ON I=ON	CACHE D=ON I=ON
Ran from ST-RAM	Ran from ST-RAM

Gembench-testi: Atari Falcon 030 (68030, 16 MHz + 68882, 16 MHz) verrattuna CT60e:hen (68060, 100 MHz).

löytyi lopulta parhaiten CT60e:n kanssa toimiva versio. Samalla tein varmuuden vuoksi myös kortin ja koneen flash-muistipiirille tallennettavat asetukset uusiksi. Tämän jälkeen Teradesk toimi ongelmitta. Työpöydän grafiikkahäiriöihin auttoi yksinkertaisesti NVDI5-ohjelmiston asentaminen. NVDI päivittää koneen käyttämään prosessoria grafiikan kopioimiseen blitterin sijasta.

Työpöytäkäyttöön CT60e-kortti toi toivottua jouhevuuutta. Sekä TOS että FreeMiNT Teradeskillä toimivat 68060:n kanssa erittäin hyvin. Työpöydän värien määränkin pystyi nostamaan helposti 256:een VGA-tilassa. Erinomainen esimerkki 68060-prosessorin tehosta on tilanne, jossa pitää tarkistaa tiedostojärjestelmät koneen käynnistyessä. 68060-prosessorilla meni testikoneessa tiedostojärjestelmän tarkistukseen alle 30 sekuntia ja

68030-prosessorilla yli viisi minuuttia. Pakettien purkaminen on toinen erittäin hyvä esimerkki 68060-prosessorin nopeudesta 68030-prosessoriin verrattuna.

CT60e:ssä on kytkin, jolla saa valittua 68030- tai 68060-prosessorin. Näin kaikki erityisesti 68030-prosessorin vaativat ohjelmat ja pelit pitäisi saada helposti toimimaan. Kortin asetuksia kuten kellotaajuutta pääsee säätämään ohjelmallisesti ohjauspaneelimuodulin (CPX) avulla.

Suurimman osan perusohjelmista pitäisi toimia suoraan 68060-prosessorin kanssa, ja joihinkin epäyhteensopiviin ohjelmiin löytyy korjauksia kyseiselle prosessorille. Varsinaisia 060-versioita ei useinkaan tavallisista ohjelmista ole, vaan ohjelmista on 000-, 020-060- ja CF- (eli Coldfire) -versiot. Demot ja muutamat käännökset peleistä tekevät

mukavan poikkeuksen ja niistä on tehty erityisiä 060-versioita.

Pelit ja demot

Monista nykyisin avoimista pelimootoreista, kuten Doom, Hexen, OpenTyrian, Duke Nukem ja OpenJazz, löytyy Atariille omat 060-versiot. Doom toimi todella hyvin 66 megahertsin kelloitetulla 68060-prosessorilla. Orionin 2010-luvulla julkaisemiin kaupallisiin seikkailupeleihin, kuten Mystiin, saa 060-päivityksiä. Esimerkiksi Philia-peliin tulee 060-version mukana huoneiden näkymän vieritys hiirtä liikuttaessa, kun aiemmin huoneista oli vain muutama staattinen kuva. Demoja on jonkin verran 68060-prosessorille. Kuuluisimpia taitavat olla The Black Lotus -ryhmän Amiga-demoista tehdyt Atari-versiot. Muutamat CT60:n omistajat jopa retostelivat foorumeilla niiden olevan parempia kuin alkuperäiset. Täytyykin myöntää, ettei Atari-versioiden ainakaan tarvitse hävetä Amiga-versioiden rinnalla. Löytyy Atariille toki ihan alkuperäistekin 060-demotuotantoa, joista hyviä esimerkkejä ovat Dead Hackers Society -ryhmän 060-demot. Myös pelejä tehtaileva Orion on ainakin Unexpected Fun -nimisen tuotoksen 68060-prosessorille tehnyt. Tietenkin useat 030-demot pyörivät myös 68060:n voimin ja toimivat toki paljon vikkelämmin kuin alkuperäisellä prosessorilla.

Kuinka nopea CT60e sitten oikeasti on?

GEMBench on ohjelma, jolla testataan usein Atarien tehokkuutta. Ohjelmaa on kehuttu jopa parhaimmaksi benchmark-ohjelmaksi. Ohjelma on alun perin ollut sharewarea, mutta myöhemmin sen koodit on avattu ja ohjelmaa on jatkokehitetty. Viimeisimmällä uudelleen kirjoitetulla versiolla tulosten pitäisi olla kiihdytetyillä koneilla tarkempia kuin aiemmillä versioilla. GEMBenchissä on myös näppärä ominaisuus, jolla voi koneiden testituloksia verrata keskenään. Laajentamattomaan kahdeksan megahertsin Atari STE -koneeseen verrattuna testikoneen CT60e-kortti on noin 33 kertaa nopeampi ja perus 030-Falconia 17 kertaa nopeampi. Yllättävää oli, että GEMBenchin liukulukutesti antoi paljon huonomman tuloksen 68060-prosessorin sisäiselle matema-

tiikkaprosessorille kuin 68030-prosessorin rinnalle liitetyle Motorolan 68882-matematiikkaprosessorille. Tes-ti saattoi tosin olla erityisesti 68882:lle kirjoitettu. Toinen suosittu ohjelma prosessorin nopeuden mittaamiseen on Kronos. Ohjelman FPU-testit CT60e:llä näyttivät neljä kertaa nopeammilta kuin matematiikkaprosessorilla varustetun 030-Falconin. Yllättävää kuitenkin oli, että Atarin kanssa yhteensopiva Hades-tietokone, joka siis on myös 68060-pohjainen, oli lähestulkoon joka testissä huonompi kuin CT60e.

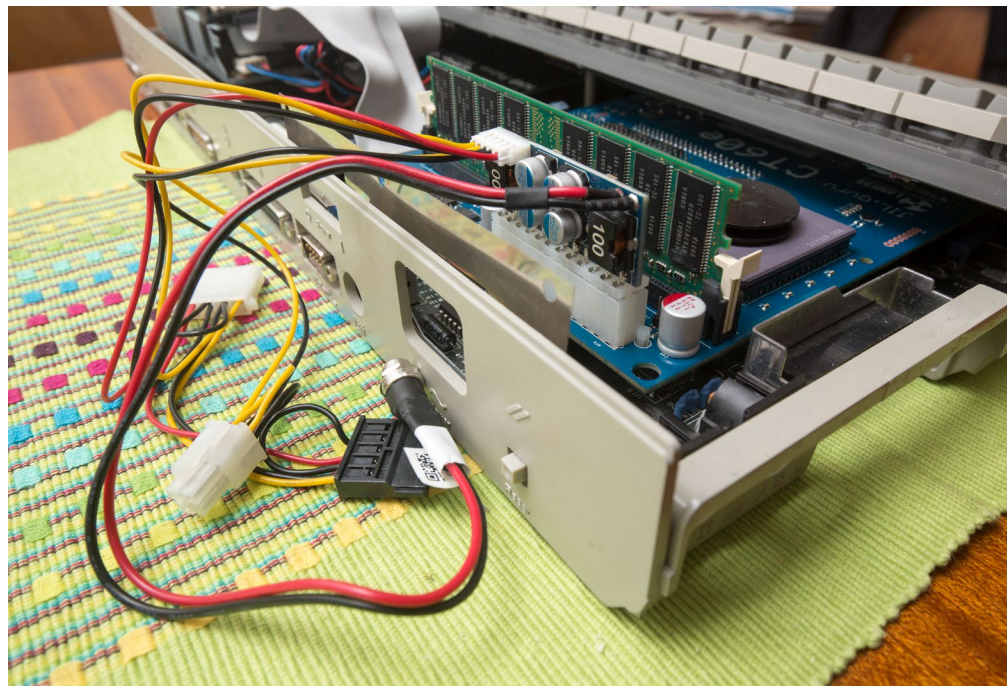
No kannattiko?

CT60e on erinomainen kortti Atari Falconin yleisen käyttömukavuuden parantamiseen. Aiemmin pelkän kookkaan ZIP-paketin purkamisenkin oli tuskallista. Ja vaikkei Falconilla CT60e-kortin voiminkaan vielä kunnolla internetiä selailla, useat tehoa vaativat pelit toimivat paremmin ja kiihdytin avaa uuden oven demojen

maailmaan. PDF-tiedostojen lukukin onnistuu, kunhan mukana ei ole kovinkaan montaa kuvaa.

Tosin harrastukset maksavat. Pelikällä kortilla oli kuitenkin hintaa 270 euroa lähipostiin kiikutettuna, ja tähän tuli vielä lisäksi 68060-prosessori, joka sekään ei mikään ilmainen ole. Muisti ei näiden päälle paljoa maksa.

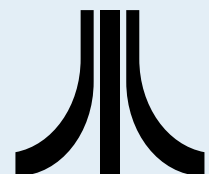
Verrattuna Amigaan Atarin käyttö 68060-prosessorin kanssa on hieman haastavampaa. Tämä selittyy varmasti osaksi sillä, että Atarissa osa ohjelmista ja peleistä toimii TOSissa ja osa FreeMiNTissä; osa ohjelmista ei toimi 68060-prosessorilla ja toiset taas vaativat 68060:n välimuistin kytkemisen pois. Amigan kanssa näistä asioista ei paljoa tarvitse huolehtia. Atarin kanssa säätämiseen kuitenkin tottuu ja käyttäessä oppii, että tietyt asiat kannattaa tehdä tietyllä tavalla. Mikäli käyttäjä on vanha konkari, nämä asiat tuskin häiritsevät. Kun huonot ja hyvät puolet lasketaan yhteen, CT60e on positiivinen kokemus. 🎮

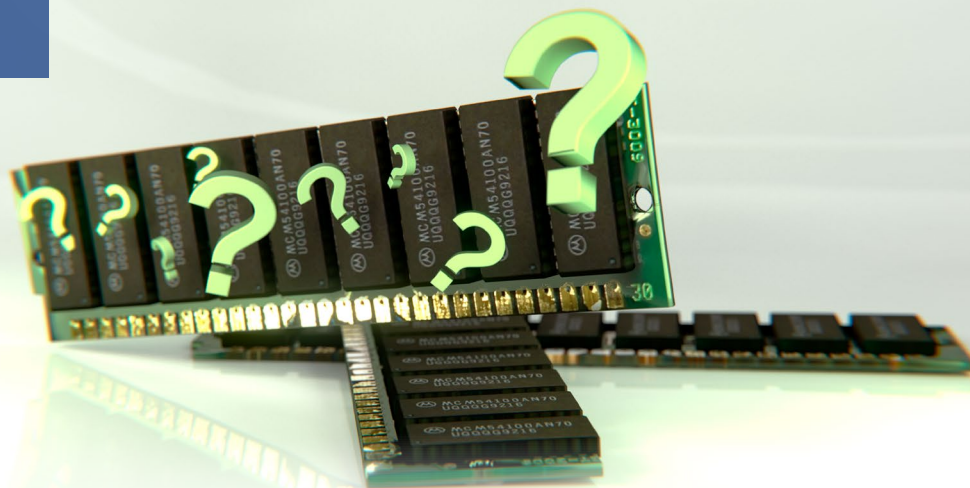


CT60e asennettuna, muisteineen ja prosessoreineen. PICO-ATX-virtalähteen liittimet ovat vielä levällään.

Lähteet

- nature.atari.org/supervidel.html
- www.atari-forum.com
- en.wikipedia.org/wiki/Atari_Falcon
- www.exxoshost.co.uk/atari/last/GB6/index.htm





Oi muistatko vielä sen **SIMM**in?

Vihjeitä muistilaatikon setvintään

Pyöriikö laatikossasi vanhoja muistikampoja, joiden tarroissa on vain jokin kryptinen mallinumero? Eikö Google löydä mallinumeron perusteella muistikamman tyyppiä?

Teksti: Ville Jouppi, kuvat: Toni Kortelahti, Ville Jouppi

SIMM-muistikammat ovat eräs yleisimmistä esineistä, joita tietokoneharrastajan laatikoihin kertyy hieman kuin itsestään. Kun niitä haluaisi käyttää jossakin, juuri oikean tyyppin ja koon löytäminen voi olla yllättävän hankalaa, sillä useimmat valmistajat eivät ole välittäneet painaa kampoihin juuri mitään tunnistetietoja. Tehtävää helpottaa, kun tuntee hieman muistipiirien ja niiden yhdistämisen perusteita.

Helpoin keino tuntemattoman muistikamman koon tunnistamiseksi on laskea siihen juotettujen muistipiirien kokojen perusteella, paljonko muistia kokonaisuudessa löytyy. Muistipiirien koot ilmoitetaan aina bitteinä, joten kun lopullinen yhteenlaskettu bittimäärä jaetaan kahdeksalla, saadaan tavuja. Muistipiirin sisäinen rakenne ilmoitetaan muodossa koko kertaa leveys, eli vaikkapa $1\text{ M} \times 4$ olisi miljoona neljän bitin levyistä sanaa, eli neljä megabittiä = 512 kilotavua. $1\text{ M} \times 1$ taas olisi miljoona bittiä yhden bitin leveydellä eli megabitti = 128 kilotavua.

30 nastaa

Vanhoista 386- ja 486-emolevyissä löytyviin 30-nastaisiin SIMM-kampoihin voi tallentaa kahdeksan bitin leveydeltä tietoa. Kampoja on sekä pariteetti-bitillä että ilman, joten kapasiteettia laskiessa täytyy olla hieman ajatusta mukana. Mikäli muistipiirejä on pariton määrä, kyseessä on pariteettinen

kampa, joita kutsutaan joskus myös yhdeksänbittisiksi.

Normaalin pariteettittoman kamman koko ratkeaa yksinkertaisesti laskemalla muistipiirien bitit yhteen ja jakamalla tulos kahdeksalla. Mikäli kammasta löytyy ylimääräinen pariteetti-piiri, täytyy tarkastella kampa hieman tarkemmin. Jos kaikki piirit ovat keskenään identtisiä, jätetään yksi piiri pois laskuista. Aivan viimeisissä 30-nastaisissa kammoissa on usein kaksi isoa ja yksi pienempi muistipiiri. Näissä tapauksissa jätetään pienin piiri huomiotta ja lasketaan yhteen ainoastaan kahden isomman piirin koot.

Otetaan esimerkiksi kampa, jossa on yhdeksän MCM54100AN70-piiriä. Mallinumeron lopusta 4100 kertoo meille, että piirin rakenne saattaisi olla $4\text{ M} \times 1$ ja lopun 70 antaa osviittaa 70 nanosekunnin hakuajasta. Pikaisen internet-haun jälkeen epäilykset vahvistuvat, ja kyseessä tosiaankin on $4\text{ M} \times 1$ kokoinen fast page mode -muisti 70 nanosekunnin hakuajalla. Piirejä on yhdeksän, joten unohdetaan se ylimääräinen piiri. $4\text{ M} \times 1$, leveys kertaa kahdeksan on $4\text{ M} \times 8$, $4\text{ M} \times 8 / 8 = 4\text{ M}$, joten kamman koko on neljä megatavua.

Toinen esimerkki on kolmepiirinen. Siitä löytyy kaksi kappaletta M514256-10J-piirejä ja yksi M41256A. Taas pariton määrä, kaksi samanlaista ja yksi erilainen piiri. Unohdetaan se erilainen suoraan, se on pienempi

pariteetti-piiri. Vauhtia on 100 ns, ja 4256 kertoo meille, että koko on suurella todennäköisyydellä $256\text{ k} \times 4$. Datalehti vahvistaa olettamuksen: $256\text{ k} \times 4$ bittiä. Leveys kertaa kaksi on $256\text{ k} \times 8$ bittiä eli 256 kt.

Viimeinen esimerkkinne sisältää vain kaksi piiriä: molemmat ovat tyyppiä MN414400ASJ-06. Nopeus on 60 ns, ja olisiko koko jälleen $4\text{ M} \times 4$? Tarkistetaan. Tässä saikin käyttää hieman luovuutta internet-haussa, mutta 414400 vaikuttaisi olevan sittenkin kooltaan $1\text{ M} \times 4$. Koska piirejä on kaksi, $1\text{ M} \times 8 / 8 = 1\text{ Mt}$, eli koko on yksi megatavu ilman paritettia.

30-nastaiset SIMMit toimivat aina viiden voltin käyttöjännitteellä, ja pariteettillisiä ja pariteettittomia voi useimmiten sekoittaa keskenään ilman ongelmia. Tunnetun poikkeuksen muodostaa Atari STE, jonka muistin laajentaminen pariteettillisilla SIMMeillä johtaa ongelmiin.

72 nastaa

72-nastainen SIMM säilöo sisällään 32 bitin leveydeltä hyötydataa, mutta muistipiirien todellinen yhteenlaskettu leveys voi virheenkorjauksesta riippuen olla myös 36, 39 tai 40 bittiä.

Näissä uudemmissa SIMM-kammoissa on lupaavan näköiset Presence Detect -nastat PD1-PD4, mutta valitettavasti niiden avulla ei voida tunnistaa muistikampaa, sillä jokaista bittikuviota vastaa neljä eri muistikampaa. Esimer-

kiksi PD1 maissa ja loput kellumassa tarkoittaa 60 ns:n hakuaikaa ja kokoja $256\text{ k} \times 36$, $16\text{ M} \times 32/36$, $4\text{ M} \times 36$ tai $256\text{ M} \times 32/36$. Joudumme jälleen las-kemaan muistipiirien kokonaisuuteen.

Otetaan esimerkiksi SIMM, jossa on kahdeksan kappaletta HM5117400BS6-piirejä. Mallinumeron loppupäästä voidaan jo suoraan päätellä, että hakuai-ka lienee 60 nanosekuntia. Muuten malli-numero ei valaise mitenkään erityisen paljon: ehkäpi piiri on neljä bittiä leveä. Hakukoneen avulla kaivetaan datalehti esiin, ja siitä käykin ilmi, että piiri on kooltaan $4\text{ M} \times 4$. Piirejä oli kahdeksan, joten yhteenlaskettu bittimäärä on $4\text{ M} \times 32$. $4\text{ M} \times 32 / 8 = 16\text{ M}$, eli ky-seessä on 16 megatavun kampa.

Toinen 72-nastainen esimerkkinne sisältää kuusitoista kappaletta MT4C-4M4E8DJ-6-piirejä. Voisi olla $4\text{ M} \times 4$ ja 60 nanosekuntia, minkä internet-haku vahvistaa. Piiri on myös EDO-tyyppi-nen. $4\text{ M} \times 64 / 8 = 32\text{ M}$, joten kyseessä on 32 megatavun EDO SIMM.

72-nastaisia muistikampoja on sekä 5 että 3,3 voltin käyttöjännitteellä. Nämä tunnistaa helposti kontaktirivien välissä olevasta puolikuun muotoisesta kuopasta. Mikäli kuoppa on kontaktirivien keskellä, käyttöjännite on 5 V. Mikäli kuoppa on sivussa ykkösnastan suun-nassa, käyttöjännite on 3,3 V. Emolevyn SIMM-kannoissa on pieni kohouma es-tämässä vääränlaisten muistikampojen asentamista emolevyyn.

ECC-muisti

Jos olet joskus purkanut vanhoja Pentium-aikaisia palvelimia, saattaa nurkissasi olla virheenkorjaavaa ECC-muistia. Mikäli nasta 48 on maissa, kyseinen kampa toimii virheenkor-jaavana ECC:tä tukevissa emolevyis-sä. Usein voit kytkeä ECC-kampasi normaaliinkin emolevyyn, jolloin vir-heenkorjaus ei vain ole käytössä. ECC-muisteissa on kammasta riippuen 36, 39 tai 40 bittiä, joten laskeminen ei aina ole suoraviivaista.

ECC-esimerkissämme on kaksi-puoleinen kampa, joka sisältää 18 kpl HYB5117405BJ-60-piirejä. Tästä voi luotettavasti päätellä vain hakuajan 60 nanosekuntia, joten haemme dataleh-den. Vastauksena on jo tylsäksi käynyt $4\text{ M} \times 4$ EDO 60 ns. Koska piirejä oli pariton määrä per puoli, pudotamme kaksi piiriä pois laskuista. $4\text{ M} \times 64 / 8 = 32\text{ M}$. Koska yhdellä puolella on yh-deksän kappaletta neljän bitin levyistä piiriä, on kyseessä 36-bittinen muisti-kampa.

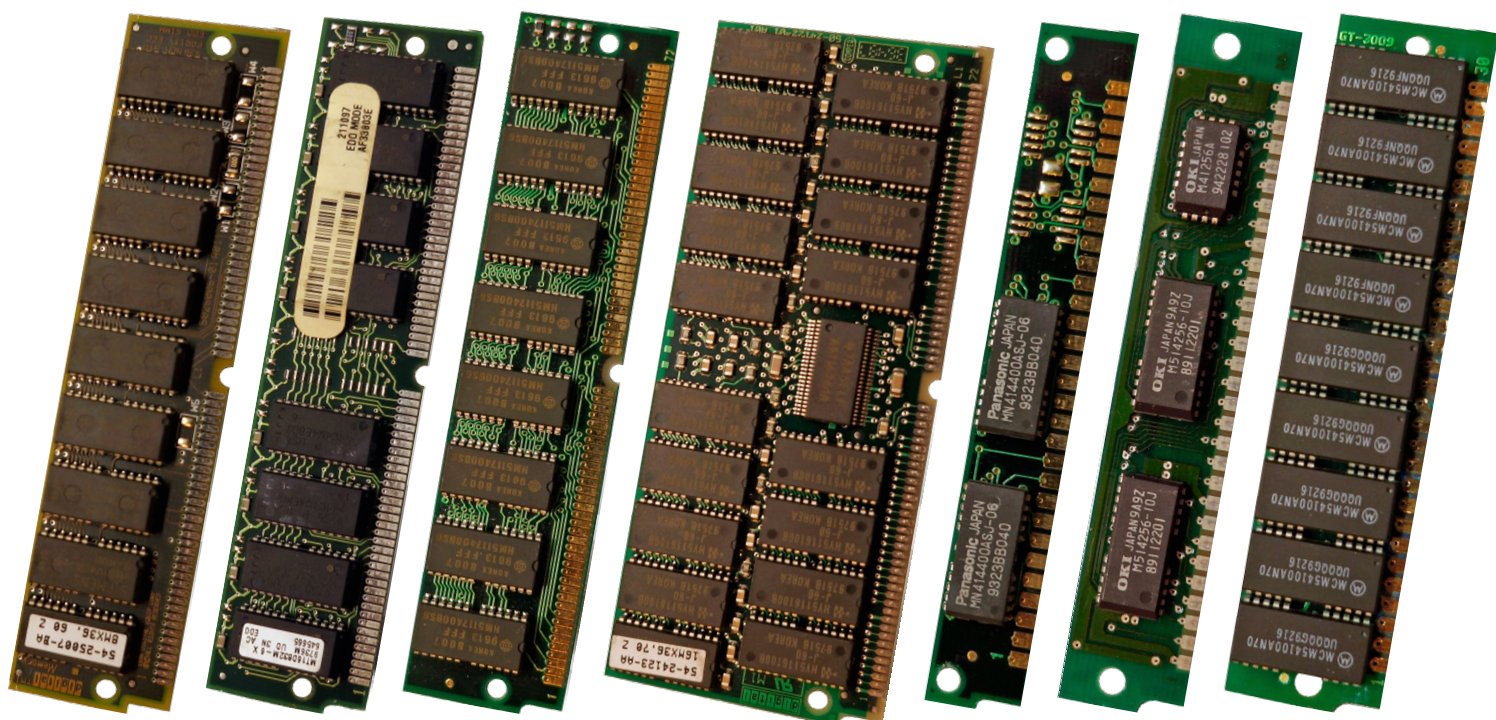
Toinen ECC-esimerkkimme on järeä. Kaksipuoliselta muistikammalta löytyy yhteensä 36 kpl HY5116100BJ-60-piiriä. Yhden piirin koko on $16\text{ M} \times 1$, eli koko kamman leveys olisi sitten $16\text{ M} \times 36$. Aiemmin määritte-lemämme nyrkkisääntö parittomasta piiriluvusta ei nyt pädekään: kammassa on 18 kpl piirejä per puoli. Onni onnettomuudessa, että piirit olivat

yhden bitin levyisiä, joten emme jou-du kovin kauaa raapimaan päättämme. Koska hyötydataa mahtuu 72-nastai-seen piiriin vain 32 bitin leveydeltä, voimme helposti vain pudottaa neljä ylimääräistä bittiä pois piirien yhteen-lasketusta leveydestä, eli $16\text{ M} \times 32 / 8 = 64\text{ M}$. 64 megatavua ja virrankulutus varmasti valtava.

Ne, jotka pääsivät karkuun

72-nastaisissa muistikammoissa käy-tetyt piirit ovat ikäluokkaa, johon löy-tyy suhteellisen helposti datalehdet internetistä. Sen sijaan 30-nastaisissa kammoissa on joskus piirejä, joiden tekniset tiedot ovat kadonneet maan päältä. Tällöin voidaan hieman päätel-lä tutkimalla muistikamman reunaan. Mikäli osoitelinjat A9 (nasta 18), A10 (nasta 19) ja A11 (nasta 24) ovat kyt-kemättä, kampa on todennäköisesti 256 kilotavua. A10 ja A11 kytkemättä viittaa yhden megatavun kampaan, ja pelkästään A11 kytkemättä on luul-tavasti neljän megatavun kokoinen. Kaikki kolme ylintä osoitelinjaa ovat kytkettyinä vain harvinaisissa 16 me-gatavun kammoissa.

SIMM-kamman koon selvittelyyn voi helposti hurahtaa hetki jos toi-nenkin. Siksi suosittelen liimaamaan muistikampaan teipin tai tarran ja kir-joittamaan siihen kyseisen kamman muistimäärän, niin säästyt toistuvilta datalehtien lukemisilta jatkossa. 🐛





EI NÄIN!

Tammerterho tipahti

Acorn Computers teki vallankumouksen suoritinmarkkinoilla, mutta jätti hyödyntämättä tilaisuuden mullistaa samalla kotimikrot.

Teksti: Mikko Heinonen

Kuvat: Mikko Heinonen, Manu Pärssinen, Wikimedia Commons, Pixabay

On oikeastaan hassua kirjoittaa *Ei näin!* -sarjassa yrityksestä, jonka tuotteiden suoria jälkeläisiä meillä lähes jokaisella on ainakin ollut taskussaan. Toisaalta juttusarjan ideana on alusta asti ollut alusta asti nostaa esiin myös hukattua potentiaalia, ja sitä Acornista ja sen tuotteista löytyy vaikka muille jakaa.

Vuonna 1978 Cambridgessa perustettu Acorn Computers pääsi kiinni leveämpään leipään, kun saarivaltakunnan yleisradioyhtiö BBC valitsi sen tarjouskilpailulla toteuttamaan tietokone-lukutaito-ohjelmaansa. Aiemmin kehitetyn Acorn Atomin pohjalta luotiin BBC Micro, josta tuli sittemmin hyvin tuttu kaikille koululaisille 1980-luvun Isossa-Britanniassa. Odotuksena oli reilun kymmentuhannen kappaleen myynti, mutta Beebiä myytiin lopulta puolitoista miljoonaa kuutena eri versiona. Lisäksi Acorn vielä teki sen pohjalta halvemman, lähinnä pelikäyttöön sopivan Electronin.

Nykyhetkessä tarkasteltuna BBC Micro erottuu aikalaisistaan edukseen etenkin monipuolisten liitäntöjen ja hyvän BASICinsa puolesta. Myös näppäimistö on asiallinen, ja syy on selvä:

koska valtio tuki mikrojen hankintaa, konetta ei tarvinnut suunnitella hinta edellä. MOS 6502-suorittimeen pohjannut arkkitehtuuri oli sekin varsin ajantasainen etenkin esittelyvuonnaan 1981. Mutta varsinainen mullistus oli vasta tulossa.

RISCi realisoituu

Tammikuun 1988 Mikrobitti teki lukijoilleen harvinaisen selväksi, mihin laitteeseen päiväunet kannattaisi jatkossa kohdistaa. Se oli nimittäin saanut unelmatestiin BBC Micron seuraajan, joka kulki nimellä Archimedes. Koneen siuksista löytyi aivan uusinta teknologiaa edustava ARM2-suoritin, jossa oli tuolloin muodikas RISC-käskykanta. Tämä tarkoittaa, että suoritin sisältää pienen määrän yksinkertaisia, nopeasti suoritettavia käskyjä. Tehokkaista MIPS- ja SPARC-työasemista tuttua tekniikkaa ei ollut koskaan aiemmin sovellettu kotitietokoneissa.

Mikrobitin testijutun kirjoittaneiden tulevan kansanedustajan ja it-miljonäärin innostus paistaa tekstistä läpi, eikä se aiheetonta olekaan. Archimedes oli järkyttävän nopea ja peittosi esimerkiksi kohtuullisen tehokkaina pidetyt

Amigan ja Atari ST:n menneen tullen. BASIC-ohjelman suorituksessa erot olivat kymmenkertaisia. Eikä grafiikka- tai äänipuolellakaan ollut valittamista, sillä VIDC1a-videopiiri tarjosi näytölle jopa 256 väriä ja tarkkuuksia, joiden esittämiseen keskimääräisen käyttäjän näyttölaite ei riittänyt vielä moneen vuoteen. Samaisen VIDC1a:n tuottamissa äänissä oli kahdeksan kanavaa Amigan neljän sijaan.

Kahden hengen show

Oikeastaan hämmästyttävintä oli kuitenkin, että Acorn oli onnistunut luomaan kokonaan uuden suoritinarkkitehtuurin. BBC Micron menestyksen jälkeen Acorn oli lähtenyt kehittämään sille laajennusta, joka sopisi myös yrityskäyttöön, mutta sopivaa ja riittävän tehokasta suoritinta ei testeistä huolimatta löytynyt. Lopulta insinöörit Steve Furber ja Sophie Wilson rohkaistuivat kokeilemaan sellaisen kehittämistä itse tutustuttuaan Berkeleyn yliopiston RISC-projektiin ja nähtyään myös, miten pieni tiimi lopulta työskenteli 6502:n parissa. Samainen piiri toimi muutenkin Acorn RISC Machinen, tuttavallisemmin ARMin, suunnittelufilosofian esikuvana.

Vuosien kehitystyön tuloksena ensimmäinen ARM asennettiin lisäkorttina vauhdittamaan BBC Microa. Seuraava tavoite oli ARMia yksinomaisesti käyttävä tietokone. Suomalaisia tietokonejournalisteja hurmasikin Archimedeksen suosiksissa jo uudempi versio ARM2.

Tulkkauksen tarvetta

Vaikka laitteen tehopusioli oli kunnossa, eräs asia oli jäänyt kehityksen tohinassa lapsipuolen asemaan – koneen käyttöjärjestelmä. Archimedes lienee yksi harvoja tietokoneita, jonka mukana toimitettu graafinen käyttöliittymä pyöri BASIC-tulkilla. Arthur-niminen viritelmä, jonka nimen huhut kertoivat olevan lyhennelmä sanoista ”A Risc operating system by THURsday”, ei ollut täysin sitä, mitä tuplasti Amiga 500:n hintaisen laitteen ostajan oli lupa odottaa. Vaikka suuri laskentateho pitkien käyttönopeuden kohtuullisena, keskeneräisyys paistoi sieltä täältä. Jopa aikalaistesteissä etenkin komentoriviä pidettiin harvinaisen kömpelönä. Myöhemmin Arthur korvattiin RISC OS:llä, joka oli jo huomattavasti elegantimpi käyttöjärjestelmä.

Tarkemmin tutkittuna koneessa oli muutamia muitakin erikoisuuksia. Vaikka sen suorituskyky oli omiaan myös vaativien pelien pyörittämiseen, laitteeseen ei ollut asennettu peliohjainporttia. Tämä paitsi vaikeutti pelien pelaamista, myös johti tilanteeseen, jossa markkinoille syntyi muutamia erilaisia ohjainstandardeja, eikä mikään peli tukenut niistä jokaista. Paljon puhutut kahdeksankanaavaiset äänet sai ulos vain joko laitteessa itsessään olevien varsin heikkolaatuisten kaiuttimien kautta tai kuulokeliitännästä, johon oli asennettu äänenlaatua heikentävä suodin.

Missä markkinointi?

Virheetöntä tietokonetta ei ole tähän päivään mennessä valmistettu, eikä Archimedes sisältänyt mitään sellaista valuvikaa, joka olisi käytännössä estänyt sitä nousemasta mukaan kilpailuun markkinaosuuksista. Näin ei kuitenkaan tapahtunut: RISC-tehohirmu jäi saarivaltakunnan erikoisuudeksi eikä koskaan myynyt sellaisia määriä, että sille olisi syntynyt merkittävää ohjelmistotuotantoa. Muutama hyötyohjelma ja peli, kuten nuotitusohjelmisto Sibelius ja aivan erityinen versio Elite-avaruuspelistä olivat niitä harvoja syitä

laitteen hankintaan. Muuten tarjolla oli enemmän tai vähemmän käännettävää, ja pelipuoille jälki näytti enemmän samanlaiselta kuin yli puolet halvemmilla Amigalla ja Atari ST:llä.

Hinta olikin ilmeisin hidastava tekijä koneen leviämiseksi. ARM-suorittimen kehitys oli ollut pitkä ja kallis operaatio, minkä lisäksi koulukoneita valmistettaessa Acorn oli todennäköisesti tottunut siihen, että monipuolisuus ja luotettavuus ovat tiukalle puristettua hintaa tärkeämpiä. Edes hieman edullisemmän ja kompaktimman A3000-mallin kehittäminen vei liki kaksi vuotta, mikä tehokkaasti söi Archimedeksen teknistä etumatkaa. Samainen A3000 oli myös viimeinen BBC-nimellä myyty mikro, joten jatkossa Acorn joutui luottamaan vain itseensä laitteiden markkinoinnissa.

Muulla Euroopassa mainontaan ei juuri panostettu. Vaikka varsin positiivinen lehtijuttu nostatti varmasti ostohaluja ainakin Suomessa, mainoksia ei näkynyt, ja maahantuojan toiminta tuntuu olleen varsin pienimuotoista. Saksassa sentään syntyi jonkinlainen harrastustoimintaa, ja käyttöjärjestelmästä tehtiin oma kieliversio maan markkinoille.

Aineksia enempäänkin

On vaikea olla spekuloimatta, mitä olisi tapahtunut, jos Acorn olisi heti aluksi laihduttanut Archimedeksestä samanlaisen halpaversion kuin Electron oli BBC Microsta. Hyvin tukevasti rakennettu ja erillisellä näppäimistöllä varustettu laitteeseen ylispektausta olisi huolelta voinut hieman keventää, jos se olisi höylännyt hinnasta puntia pois. Kun A3000 saapui, se oli askel oikeaan suuntaan, mutta edelleen melko hintava ja jo hieman myöhässä.

Tällaisenaan Archimedes jäi pahemman kerran muna ja kana -ongelman uhriksi. Muu kuin opetuskäyttöön tarkoitettu ohjelmisto oli vähäistä, mikä ei lisännyt koneen houkuttelevuutta. Ja koska softalle ei ollut ostajia, sitä ei kannattanut tehdä. Pelien käännökset 16-bittisiltä kotimikroilta olivat keskimäärin kauniita, mutta eivät tarjonneet

juuri mitään lisää alkuperäisiin versioihin nähden. Sillä oli vaikea perustella tuplahintaisen koneen hankintaa.

Laitteiston kehitys joka tapauksessa jatkui, joskin melko hitaasti. Vuonna 1991 saatiin ensimmäiset ARM3-suorittimeen perustuvat mallit ja seuraavana vuonna ARM3:n halvempi versio ARM250. Viimeiset Archimedes-mallit julkaistiin 1992, minkä jälkeen Acorn siirtyi RiscPC-arkkitehtuuriin. Se pyöri Archimedeksen tavoin RISC OS:llä ja eli peräti vuoteen 2003 asti. Acorn itse luopui kuitenkin valmistuksesta jo 1990-luvun lopussa, minkä jälkeen puuhaa jatkoi Castle Technology. Viimeiset RiscPC:t olivat StrongARM-suorittimeen perustuvia numeromurskaimia, mutta siinä vaiheessa muu maailma oli jo valinnut työpöydälleen joko Intelin x86:n tai Motorolan, Applen ja IBM:n PowerPC:n.

Käsivarret taskuissa

Acornia Archimedeksen maailmanvalloituksen epäonnistuminen tuskin ihan hirveästi harmitti, sillä koneen sisukissa makasi varsinainen kultakaivos. ARM-arkkitehtuurilla oli kilpailijoihinsa verrattuna se ylivoimainen etu, että se oli todella virtapihi. Teknologia kiinnostasi mobiililaitteiden kehittäjiä, jotka halusivat ostaa sen valmistusoikeuksia. Jo vuonna 1990 ARM-kehitys irrotettiin omaan yhtiönsä, jolloin ARM-lyhenne muutettiin tarkoittamaan ”Advanced RISC Machines”.

Ensimmäiset valmistuslisenssin hankkijat ilmaantivat nopeasti ARMin ovelle, ja nykyisin yli miljardin liikevaihtoa pyörittävä yhtiö ilmoittaa markkinaosuutensa älypuhelimien suorittimista olevan vaatimatonta 95 %. Kolme miljardia ihmistä

käyttää ARM-suorittimella varustettua laitetta päivittäin.

Yhden piirilevyn tietokone Raspberry Pi on sekin ARM-pohjainen ja lähtöisin brittiläisen säätösuunnittelupöydältä. On siis jokseenkin sopivaa, että sillä voi ajaa – ilmaiseksi ja tehokkaasti – RISC OS:ää ja BBC BASICia. 🐉




```

static long DoStructInternal (long Offs, unsigned Type) /* Handle the .STRUCT command */ { long Size = 0; /* Outside of other structs, we need a name. Inside another
struct or * union, the struct may be anonymous, in which case no new lexical level * is started. */ int Anon = (CurTok.Tok != TOK_IDENT); if (!Anon) { /* Enter a new
scope, then skip the name */ SymEnterLevel (&CurTok.SVal, SCOPE_STRUCT, ADDR_SIZE_ABS, 0); NextTok (); /* Start at zero offset in the new scope */ Offs = 0; } /*
Next for end of line */ ConsumeSep (); /* Read until end of struct */
while (CurTok.Tok != TOK_ENDSTRUCT && CurTok.Tok != TOK_ENDUNION && CurTok.Tok != TOK_
and comment lines */ if (CurTok.Tok == TOK_SEP) { NextTok (); continue; } /* The format
(CurTok.Tok == TOK_IDENT) { /* Beware: An identifier may also be a macro, in which
(M) { MacExpandStart (M); continue; } /* We have an identifier, generate a symbol
symbol the offset of the current member */ SymDef (Sym, GenLiteralExpr (Offs),
Read storage allocators */ MemberSize = 0; /* In case of errors, use zero */
break; case TOK_DBYT: case TOK_WORD: case TOK_ADDR: NextTok (); MemberSize =
(3); break; case TOK_DWORD: NextTok (); MemberSize = Member (4); break; case
is missing"); } else { MemberSize = Member (1); } break; case TOK_TAG: NextTok
("Unknown struct/union"); } else if (GetSymTabType (Struct) != SCOPE_STRUCT)
= GetSizeOfScope (Struct); if (!SymIsDef (SizeSym) || !SymIsConst (SizeSym,
MemberSize = Member (MemberSize); break; case TOK_STRUCT: NextTok (); MemberSize =
MemberSize = DoStructInternal (Offs, UNION); break; default: if (!CheckConditionals
allocator in struct/union"); } } /* Assign the size to the member if it has a name
(Type == STRUCT) { /* Struct */ Offs += MemberSize; Size += MemberSize; } else {
end of line */ ConsumeSep (); } /* If this is not a anon struct, enter a special
holds the size of the * struct. Since the symbol starts with a dot, it cannot
(!Anon) { /* Add a symbol */ SymEntry* SizeSym = GetSizeOfScope (CurrentScope);
NONE); /* Close the struct scope */ SymLeaveLevel (); } /* If this is not a
table of the struct that holds the size of the * struct. Since the symbol
the struct scope level. */ if (!Anon) { /* Add a symbol */ SymEntry* SizeSym
(Size), ADDR_SIZE_DEFAULT, SF_NONE); /* Close the struct scope */ SymLeaveLevel (); }
(TOK_ENDSTRUCT, "", ENDSTRUCT' expected"); } else { Consume (TOK_ENDUNION, "", ENDUNION'
GetStructSize (SymTable* Struct) /* Get the size of a struct or union */ { SymEntry*
struct/union is unknown"); return 0; } else { return GetSymVal (SizeSym); } } void
STRUCT); } void DoUnion (void) /* Handle the .UNION command */ { DoStructInternal (0,

```

TAUON paikka!

Teksti: Markus Kuula
Kuvat: Nasu Viljanmaa,
Markus Kuula

Tietokoneella työskenneltäessä aika kuluu huomaamatta ja tarpeellisten verryttelytaukojen pitäminen voi unohtua. Seuraa-vaksi esiteltävä Windows-ohjelma korjaa tämän puutteen.

Intensiivinen koneella työskentely saattaa viedä ajantajun, joten jonkun olisi tasaisin väliajoin muistuttava tauon pitämisestä. Koska tällaisen henkilön palkkaaminen tulisi kalliiksi, eivätkä yritykset välttämättä suvaitse ulkopuolisia henkilöitä tiloissaan, muistuttajan olisi oltava kone. Valjastetaan siis tietokone muistuttamaan tauoista.

Tauottajan määrittely

Tauutusohjelma kehottaa tietokoneen ääressä olijaa pitämään venyttelytauoja. Kehotus annetaan aina puolen tunnin välein. Vanhoissa liikuntasuosituksissa väli on saattanut olla pidempikin: tunti

tai jopa 2 tuntia. Pitäydytään silti tiheämmässä muistutusväliä.

Ohjelman itsensä voi myös asettaa tauolle. Tällöin se ei häiritse käyttäjää, jos hän on esimerkiksi esittämässä muille näyttönsä sisältöä. Muuten varsinaista säädettävää ei tarvita, eikä näin ollen kummoista käyttöliittymääkään.

Pienenä lisäominaisuutena laitetaan ohjelma muistuttamaan koneen ääressä olijaa uudelleen, tällä kertaa 10 minuutin välein, jos hän ei nosta ahteriaan tuolilta.

Tekniikka

Kehotus voidaan toteuttaa vaikkapa perinteisellä viesti-ikkunalla, joka pomp-sahtaa keskelle ruutua. Tällöin se tulee takuuvarmasti huomatuksi, mutta samalla keskittyminen työn alla olevaan asiaan herpaantuu. Raporttia kirjoittava työntekijä ei välttämättä ilahdu, jos kesken kirjoituksen näytölle tulee ruutu, jonka vuoksi osa näppäinpainalluksista hukkuu. Näin voi kuitenkin tapahtua, jos tauottajan viestiruutu ryhtyy yhtäkkiä aktiiviseksi sovellukseksi.

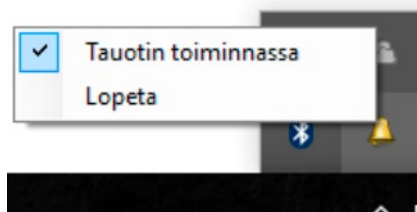
Parempi valinta onkin muistuttaa käyttäjää tauosta häntä häiritsemättä, mutta kuitenkin huomiota herättävästi. Tämä käy päinsä, kun käytämme kuplamuotoista viestiä. Puhekuplaa muistuttavia viestejä voivat näyttää ruudulla ne ohjelmat, jotka ovat tehtäväpalkin niin sanotulla ilmoitusalueella. Tämä alue sijaitsee heti kellon vieressä, yleensä alaoikealla.

Ilmoitusalue (notification area), jota ohjelmoijat nimittävät myös järjestelmätarjottimeksi (system tray), mahdollistaa ohjelman toimimisen taustalla. Ilmoitusalueelle lisätyt ohjelmat näkyvät pieninä kuvakkeina, ilman otsikkoa.

Windows-ohjelma voidaan käynnistää periaatteessa kolmella eri tavalla: tavallisesti se käynnistyyessään ilmestyy ruudulle ja tehtäväpalkkiin. Tällöin ohjelmaan voidaan siirtyä toisesta ohjelmasta Alt+Tab-näppäinyhdistelmällä. Toisella tavalla ohjelma ei avaa ruudulle ikkunaa. Tällöin sen näkee vain Tehtävienhallinnan kohdasta "Prosessit". Se ei siis löydy kohdasta "Ohjelmat". Huonona puolena ohjelman hallinta kunnollisesti on tällöin mahdotonta, sillä käyttöliittymän puuttuessa ei ohjelmaa esimerkiksi voi helposti sulkea.

Parempi tapa onkin käynnistää ohjelma ilmoitusalueelle, jolloin käyttäjä voi hiirellä tai näppäimistöllä osoittaa ohjelman kuvaketta ja avata pikavalikon. Monesti ensimmäinen valinta avaa ohjelman ruudulle. Meidän ohjelmamme ei kuitenkaan tee niin, sillä tarvitsemme vain kaksi kohtaa: ohjelman kytkeminen pois tai päälle ja sen lopettaminen. Meidän on kyllä luotava ikkuna, mutta voimme jättää sen piiloon.

Kaikkien Windows-ohjelmien on luotava ainakin yksi ikkuna, jotta ne voivat toimia ja kommunikoida järjestelmän kanssa. Mikäli ohjelmaamme kuuluisi enemmän asetuksia, ne



olisi järkevää koota ikkunaan, jonka käyttäjä voi pikavalikon kautta avata. Tällöin erillistä piiloikkunaa ei tarvittaisi. Asetusruutu olisi tällöin olemassa piilotettuna ja otettaisiin esille vain tarpeen vaatiessa.

Vielä pitäisi ratkaista eräs asia: kuinka ohjelma tietää, onko käyttäjä koneen äärellä vai ei? Yksinkertaisin ajatus olisi ohjelmoida se hälyttämään aina tietyllä kellonlyömällä. Se kuitenkin rajaa pois liukuvaa työaikaa noudattavat työpaikat, eikä ota muutenkaan huomioon päivän tapahtumia. Kuittausnäppäin toimisi paremmin, mutta se vaatii käyttäjää muistamaan asian. Paras ratkaisu onkin luottaa asiaan, jonka toivon mukaan jokainen tekee ennen kuin poistuu koneen ääreltä: lukitsee Windowsin. Kun istunto lukitaan, ohjelma pysäyttää ajastimen ja käynnistää sen uudelleen kun lukitus avataan.

Toteutus

Toteutuskielenä on C# ja kehitysympäristönä Visual Studio 2013 Community Edition, mutta ohjelma kääntyy myös ainakin 2015-versiolla. Aloita uusi projekti ja anna sen nimeksi vaikka Tauottaja. Projektin .NET-versiona käy 2. Muista valita Windows Application, sillä ohjelmassa tarvitaan tyhjä lomake.

Ikkunan piilotus tehdään määrittämällä se message only -tyyppiseksi. Tällaisella ikkunalla ei ole näkyviä osia, vaan se ainoastaan kykenee lähettämään ja vastaanottamaan viestejä käyttöjärjestelmältä ja toisilta ohjelmilta.

Rivillä 34 määritellään käsittelijä, joka suoritetaan, kun istunto lukitaan tai lukitus avataan. Sen toisena parametrina on tieto, mitä istunnolle tapahtuu. Jos tietokoneen – ja tuolin – käyttäjä ei lukitse istuntoa 30 minuuttiin, tulee viesti ilmoitusalueelta. Samalla ajastimen tikitysväliä tihennetään siten, että seuraava viesti tulee 10 minuutin päästä. Näin jatketaan kunnes käyttäjä lukitsee istunnon uudelleen, kirjautuu ulos tai sammuttaa tietokoneen.

Ohjelman käyttöliittymänä on kaksi valintaa sisältävä valikko: toisella ohjelman saa väliaikaisesti tauolle ja toisesta se lopetetaan kokonaan. Ajastimen interval-arvo säilyy muuttumattomana kun ohjelma laitetaan tauolle. Jos on tarpeen, sen voi nollata rivin 72 jälkeen.

Ohjelmassa käytettävät komponentit on lueteltu alla.

Komponentti	Nimi	Lisätiedot
ajastin	timer	pääajastin
toinen ajastin	helpTimer	aputekstin päivitys
ilmoitusalueen hallinta	notifylcon	-
pikavalikko	contextMenuStripv	-

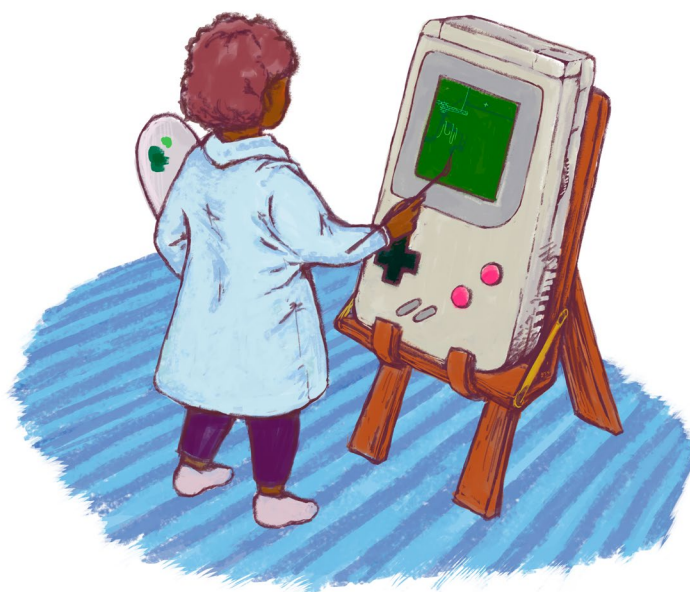
Mikäli ohjelmoiminen maistuu puulta, voi valmiin version kopioida osoitteesta

www.mwritings.net/compold/2015/11/15/have-a-break. 🐞

KOODIA

Game Boyle kotikonstein

Pelkästä pelikoneesta demoalustaksi



Teksti: Samuli Vuorinen

Kuvat: Mitol Meerna, Samuli Vuorinen, Wikimedia Commons

Useimmille 1980-luvun lapsille Nintendo Game Boy on tuttu pelikonsoli, joka kulki aina reissussa mukana. Omien ohjelmien tai pelien kehittäminen Game Boylelle oli harrastajille pitkään liki mahdotonta, sillä laite hyväksyi vain Nintendon lisensoimia pelikasetteja. Nykyään on kuitenkin saatavilla työkaluja, joilla harrastajatkin voivat kehittää omia ohjelmiaan tälle konsoliklassikolle!

Vuonna 1989 julkaistu Game Boy tarjoaa nykypäivän mittapuulla varsin vaatimattomat puitteet ohjelmien suorittamiselle. Laitteen sydän on 4 megahertsin taajuudella pyörivä, Sharpin valmistama 8-bittinen LR35902-suoritin, joka on Nintendon oma muunnelmä aikanaan yleisestä Zilog Z80 -suorittimesta. Siitä löytyvät 8-bittiset rekisterit A, B, C, D, E, H ja L. Rekistereitä voi myös yhdistellä 16-bittisiksi rekisteripareiksi BC, DE ja HL.

Laitteessa on 8 kilotavua sekä näyttö- että käyttömuistia, ja itse pelikasetilla on vähintään 32 kilotavua lukumuistia. Mitään muuta pelikasetilla ei välttämättä ole, mutta se voi sisältää ylimääräistä käyttömuistia siltä varalta, että laitteen oma käyttömuisti ei riitä.

Niin käyttö-, luku- kuin näyttömuistillekin on varattu omat alueensa laitteen muistiavaruudessa. Game Boyssa on 16-bittinen muistiväylä, joten muistiavaruuden koko on 64 kilotavua. Koska pelikasetilla voi olla suurimmillaan jopa 8 megatavua lukumuistia ja 128 kilotavua ylimääräistä käyttömuistia, 16-bittinen muistiväylä ei pysty antamaan kaikelle muistille osoitteita kerralla. Luku- ja käyttömuisti on jaettu muistipankeihin, joita voidaan ottaa käyttöön yksi kerrallaan.

Pelikasetilla olevan lukumuistin 16 ensimmäistä kilotavua ovat aina käytettävissä muistiavaruuden alussa. Muistiavaruuden seuraavat 16 kilotavua ovat valittavissa lukumuistin jäljelle jääneistä muistipankeista. Muistiavaruuden ensimmäinen puolisko on siis kokonaan varattu pelikasetin lukumuistille. Toisen puoliskon kautta käsitellään muun muassa video- ja käyttömuistia.

Game Boyssa on myös äänipiiri, josta on myöhemmin tullut varsin suosittu chiptune-muusikoiden keskuudessa. Äänipiirissä on neljä kanavaa, joista kaksi ensimmäistä tuottavat kanttaaltoja, kolmas 4-bittisiä ääninäytteitä ja neljäs kohinaa.

Vihreän eri sävyjä

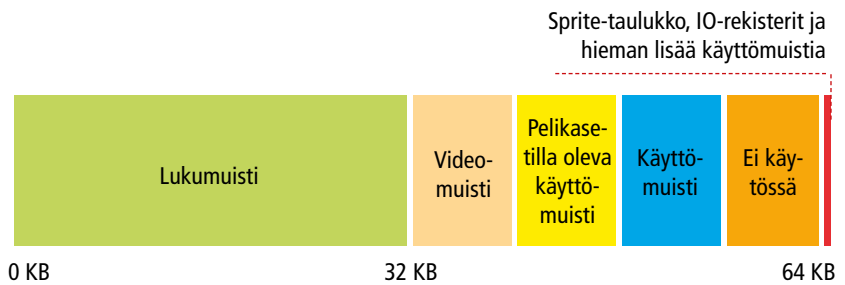
Moni muistaa Game Boy'n erityisesti siitä, miten huonosti laitteen nestekidenäytöstä sai selvää. Näytössä ei ole omaa taustavaloa, joten laitetta ei voi sellaisenaan käyttää hämärässä. Internetissä on nykyään tilattavissa jälkiasennettavia taustavaloja ja muita parannuksia Game Boy'n näyttöön.

Näyttö on 2-bittinen, joten jokainen pikseli voi näyttää 4 eri vihreän sävyä tummasta vaaleaan. Pikseleitä on yhteensä 160×144 kappaletta. Pikselien vähäinen määrä ei juurikaan haittaa, sillä näyttö on fyysisesti pieni.

Piirtäminen laitteen näytölle ei valitettavasti ole erityisen suoraviivaista. Näyttömuistiin tallennetaan 8×8 pikselin ruutuja, joihin viitataan erillisessä ruutukartassa, joka määrää mitä ruudulle piirretään. Game Boyssa on



USB-pelikasetille voi siirtää pelejä ja ohjelmia suoraan tietokoneelta.



Käyttö- ja lukumuistiin, videomuistiin sekä kaikkiin oheislaitteisiin ja ohjaimiin pääsee käsiksi muistiavaruuden kautta.

kaksi erillistä ruutukarttaa: yksi taustaa varten ja toinen taustan päälle piirrettävää tasoa varten.

Taustaa ohjaava ruutukartta käsittää 32×32 ruutua, mutta ruudulle mahtuu kerrallaan vain 20×18 ruutua. Laitteessa onkin kaksi muistipaikkaa, joihin kirjoittamalla taustakuvaa voi vierittää pysty- ja vaakasuunnassa. Taustakuvan päälle piirrettävää tasoa ei voi vierittää, mutta sen voi määrätä piirrettäväksi vain tiettyyn osaan ruutua. Tätä käytetään peleissä usein rajaamaan näytölle alue, josta pelaaja näkee esimerkiksi pelihahmon terveydentilan tai pistetilanteen.

Kahden ruutukartan lisäksi ruudulle voi piirtää maksimissaan 40 sprite-hahmoa (kuitenkin vain 10 hahmoa per pikselirivi). Sprite-hahmot ovat käytännössä kolmivärisiä, sillä yksi väri on aina läpinäkyvä. Hahmot voivat olla joko 8×8 tai 8×16 pikselin kokoisia. Molempia kokoja ei voi käyttää yhtä aikaa, vaan kaikki yhtä aikaa näkyvät sprite-hahmot ovat samankokoisia.

Sprite-hahmot voi vapaasti asemoida ruudulle, ja ne voi myös peilata pysty- ja vaak-akselin ympäri. Näyttömuistiin ei siis tarvitse tallentaa erilisiä ruutuja esimerkiksi pelihahmon eri kävelysuunnille.

Näytön käyttämistä mutkistaa entisestään se, että näyttömuistia ei voi käyttää milloin tahansa, vaan ainoastaan silloin, kun näytölle piirtäminen ei ole kesken. Jokaisen pikselirivin välissä on niin kutsuttu H-blank-jakso, jolloin näyttömuisti on käytettävissä. Vastaavasti jokaisen erillisen kokonaan piirretyn näytöllisen välillä on niin kutsuttu V-blank-jakso, jolloin näyttömuisti on myös käytettävissä.

H-blank kestää noin 200 kellosykliä ja V-blank 4 560 kellosykliä. Näyttö päivittyy noin 60 kertaa sekunnissa. Ennen näyttömuistin lukemista tai sinne kirjoittamista on siis varmistettava, että jompikumpi näistä jaksoista on menossa, tai tarpeen tullen odottaa.

Pelikasetin salat

Game Boyssa on pieni 256 tavun lukumuisti, jonka sisältämä ohjelma suoritetaan aina kun laite käynnistetään. Ohjelma yrittää pelikasetin lukumuistissa olevien otsaketietojen perusteella määrittää, onko kasetti Nintendon lisensoima. Jos otsaketiedot eivät täsmää, laite ei jatka suoritusta. Game Boyn salainen käynnistysohjelma onnistuttiin saamaan laitteesta ulos vasta vuonna 2003.

Jotta laite hyväksyy pelikasetin, kasetin muistissa tulee olla kopio Nintendo-logosta, ja sen täytyy bitti bitiltä täsmätä Game Boyn omassa muistissa olevaan logoon. Lisäksi otsaketietojen lopussa tulee olla tietyllä tavalla laskettu tarkistussumma otsaketietojen sisällöstä. Koska laite lukee Nintendon logon pelikasetilta käynnistyksen aikana, Nintendon logon tilalla näkyy pelkkä musta laatikko, jos laitteen käynnistää ilman pelikasettia.

Nyt kun muistipiirin sisältö on tiedossa, voidaan tehdä pelikasetteja, jotka laite hyväksyy mukisematta. Nykyään nettikaupoista löytyykin monenlaisia pelikasetteja, joihin voi ladata pelejä ja ohjelmia tietokoneelta USB-yhteyden kautta. Tällaisia kasetteja ovat esimerkiksi EverDrive GB, Kitsch-Bent USB 64M Smart Card tai Drag'n'Derp Cart.

Työkaluja harrastajalle

Tällä hetkellä luultavasti suoraviivaisin tapa kirjoittaa ohjelmia Game Boylle on kirjoittaa assemblya, eli symbolista konekieltä. On olemassa Game Boylle suunnattuja C-kääntäjiä, kuten GBDK, mutta ne tuottavat usein melko tehotonta ja tilaa vievää koodia verrattuna käsintehtyyn assembly-koodiin. Lisäksi GBDK:n kehitys on loppunut jo vuonna 2001.

Tällä hetkellä suosituin assembler-ohjelma Game Boyta varten on Rednex Game Boy Development System eli RGBDS. Ohjelman kehitys oli pitkään tauolla, mutta kehitystyö on jälleen nytkähtänyt liikkeelle kuluneen vuoden aikana. Kannattaa kuitenkin varautua siihen, että työkalujen dokumentaatio ei ole ajan tasalla.

RGBDS-paketin työkalut huolehtivat symbolisen konekielen kääntämisestä suorittimen ymmärtämäksi konekieleksi, käännösyksiköiden yhteenlinkittämisestä sekä otsaketietoi-

hin kirjoitettavien tarkistussummien laskemisesta.

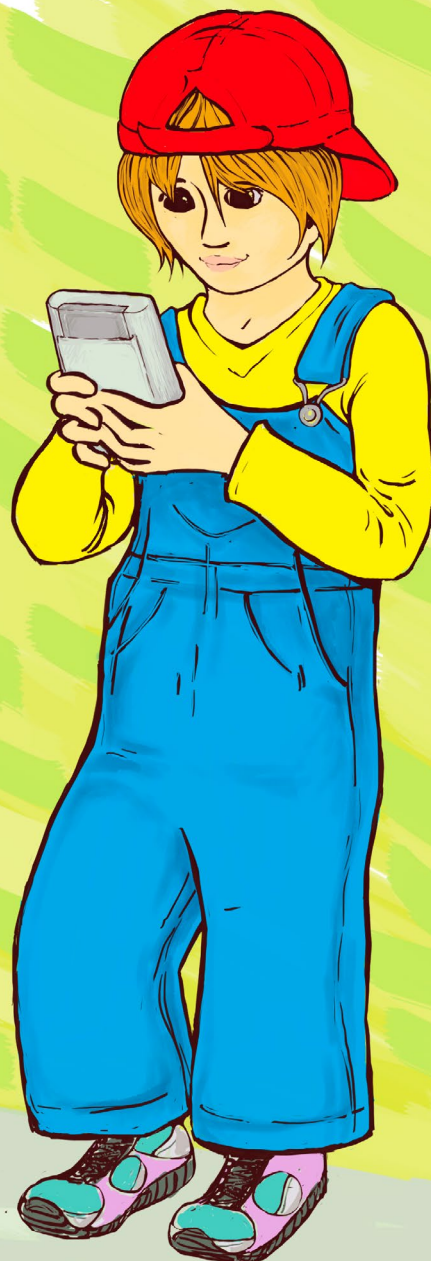
Kääntäjä tuottaa tiedoston, jonka sisältö voidaan kirjoittaa suoraan pelikasetin lukumuistiin. Kehitystyötä varten on kuitenkin helpointa käyttää emulaattoria ohjelmien ja pelien ajamiseen. Tällä hetkellä suosituin ja tarkin emulaattori on BGB, joka tarjoaa lisäksi paljon hyödyllisiä työkaluja omien ohjelmien kehittämiseen. Ohjelmia on mahdollista ajaa yksi käsky kerrallaan ja tutkia, miten esimerkiksi suorittimen rekisterien arvot muuttuvat ajon aikana tai millaisia kuvia näyttömuistiin on milloinkin ladattu.

Artikkelin lopusta löytyy hyödyllisiä linkkejä aloittelevalle kehittäjälle.

Yksi kattavimmista dokumenttikoelmista Game Boy -kehittäjälle on niin kutsuttu Pan Docs, joka sisältää hyvin yksityiskohtaiset selostukset laitteen ohjausrekistereistä, ajoituksista ja muusta käyttäytymisestä.

Oheiseen koodilistaukseen on kirjoitettu lyhyt assembly-ohjelma, joka piirtää Game Boyn näytön täyteen hymiöitä. Tilan säästämiseksi otsaketiedot on jätetty pois, mutta koko ohjelmakoodi on ladattavissa osoitteesta skrolli.fi/2017.2.

Ymmärtämällä käskyt `ld` sekä `jp` pääsee jo pitkälle. Ensimmäinen on lyhenne sanasta `load`, ja käskyllä kopioidaan dataa suorittimen rekisterien tai muistiosoitteiden välillä. Jälkimmäi-



nen käsky on hyppykäsky, eli jump, jolla voidaan ehdollisesti tai ehdottomasti siirtyä suorittamaan koodia eri paikkaan. Load-käskyn variantti **ldi** lataa annetun rekisterin sisällön HL-rekisterin osoittamaan muistipaikkaan ja kasvattaa HL-rekisterin arvoa yhdellä. 🐛

Linkejä Game Boy-kehitykseen

BGB-emulaattori

bgb.bircd.org

Rednex Game Boy Development System

github.com/rednex/rgbds

Game Boy Developer's Kit

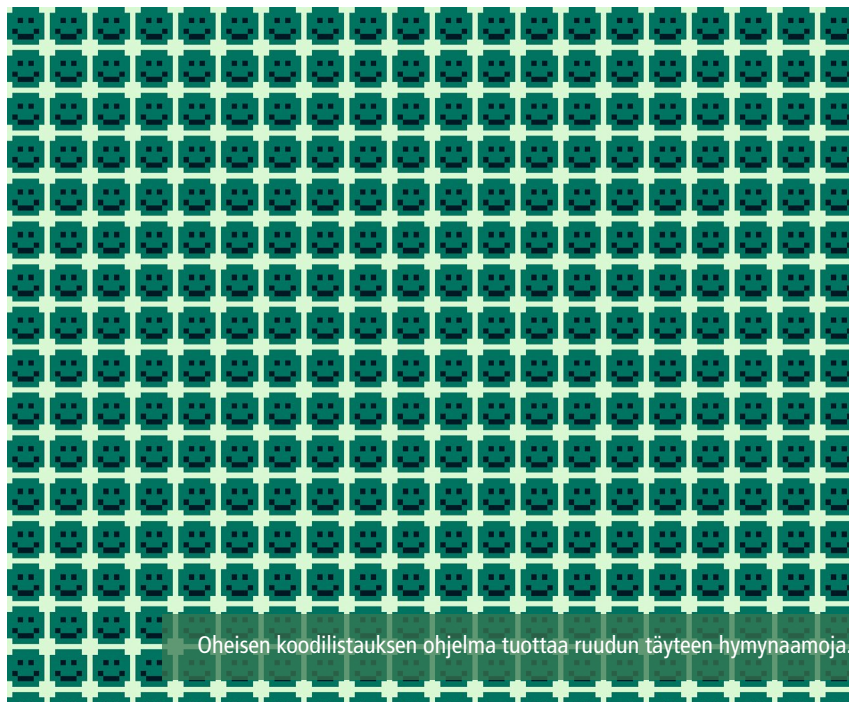
gbdk.sourceforge.net

Pan Docs

gbdev.gg8.se/wiki/articles/Pan_Docs

Awesome Game Boy Development

github.com/avivace/awesome-gbdev



Oheisen koodilistauksen ohjelma tuottaa ruudun täyteen hymynaamoja.

```
; makrot.inc-tiedostossa määritellään OdotaVBlankinAlkua-makro.
INCLUDE "makrot.inc"
```

```
; Ohjelman suoritus alkaa osoitteesta $100,
; josta hypätään osoitteeseen $150, koska otsaketiedot
; sijaitsevat välillä $104-$14F.
```

```
Section "Alku",HOME[$100]
```

```
jp aloitus
```

```
Section "Ajettava ohjelma",HOME[$150]
```

```
aloitus:
```

```
; Näyttö sammutetaan, jotta ruutudatan
; ja -karttojen latauksessa ei tarvitse miettiä
; voiko näyttömuistiin kirjoittaa.
; Näyttö voidaan sammuttaa vain V-blankin aikana,
; jonka odottamiseen on luotu makro.
```

```
OdotaVBlankinAlkua
```

```
ld A, %00000000
```

```
ld [$FF40], A
```

```
; Taustan käyttämä väripaletti valitaan kirjoittamalla
; se muistiosoitteeseen $FF47.
```

```
ld A, %11100100
```

```
ld [$FF47], A
```

```
; Näyttömuistiin ladataan 16 tavua ruutudataa
```

```
ld D, 16
```

```
ld BC, ruutu_data
```

```
ld HL, $8000
```

```
.ruudun_lataus:
```

```
ld A, [BC]
```

```
inc BC
```

```
ldi [HL], A
```

```
dec D
```

```
jp nz, .ruudun_lataus
```

```
; Osoitteessa $9800 sijaitseva ruutukartta
; täytetään ruudulla numero 0.
```

```
ld HL, $9800
```

```
ld B, 180
```

```
.ruutukartan_lataus:
```

```
ld A, 0
```

```
ldi [HL], A
```

```
ldi [HL], A
```

```
dec B
```

```
jp nz, .ruutukartan_lataus
```

```
; Näyttö käynnistetään ja taustan
; piirto laitetaan päälle.
```

```
ld A, %10010001
```

```
ld [$FF40], A
```

```
.lopetus:
```

```
halt
```

```
jp .lopetus
```

```
; Hymiötä esittävä 8x8 pikselin ruutu,
; jossa numerot 0-3 vastaavat sävyjä
; vaaleanvihreästä tummanvihreään.
```

```
ruutu_data:
```

```
DW `01111100
```

```
DW `11111110
```

```
DW `11313110
```

```
DW `11111110
```

```
DW `13111310
```

```
DW `11333110
```

```
DW `01111100
```

```
DW `00000000
```





Peliklassikko BASICilla: **STAR TREK**

Klassista Star Trek -avaruuspeliä on pelattu lukemattomissa time sharing- eli osituskäyttötietokoneissa aina 1970-luvun alusta lähtien. Siitä on tehty myös runsaasti käännöksiä kotitietokoneille. Tässä artikkelissa tutustumme pelin sielunelämään.

Teksti: Antti Ylikoski Kuva: Mikael Heikkanen

Sain idean kääntää Star Trekin modernille PC-BASICille. Tuloksena oli avaruuspelejä, jossa ei ole modernia huippugrafiikkaa, mutta joka on mukava retro-ohjelmointikokemus. Kun pelistä on olemassa täydelliset lähdekoodit, käyttäjä voi itse ohjelmoida siitä täysin haluamansa. Artikkelin käsittelemä peli on lukuisista Star Trek -peleistä kehittynein, Super Star Trek. Sen pelaaminen on hauska kokemus, erityisesti kun ohjelmaa on säädetty niin, että pelin voittaminen ei ole turhauttavan vaikeaa.

Pelin juoni

Pelin päähenkilö ja pelaaja on avaruusalus USS Enterprisen kapteeni. Klingonien avaruuslaivasto on hyökännyt kotigalaksiimme. USS Enterprisen ja sen kapteenin tehtävä on tuhota viholliset, ennen kuin ne pääsevät iskemään avaruuslaivaston päätukikohtaan. Tähän on käytettävissä tietty määrä tähti-

päiviä (stardate).

Enterprisella on kaksi asetta: vaiheiset (phasers) sekä fotonitorpedot, joilla voi tuhota vihollisia. Galaksissa on tietty joukko omia tukikohtia, joissa voi tankata energiaa sekä ladata lisää aseita ja korjata USS Enterprisen vaurioita.

Kapteenin tehtävänä on partioida ympäri galaksia ja ampuu viholliset hengiltä. Vihollisten etsimistä varten Enterprisellä on kaksi avarustutkaa: Long Range Scanner, jolla näkee muuttaman kvadrantin ympäriinsä, sekä Short Range Scanner, jolla näkee oman kvadrantin kaikki oliot. Lisäksi on olemassa tietokone, jolla saa näkyviin muun muassa erilaisia näyttöjä vallitsevasta tilanteesta.

Loput piirteet selviävät pelaamalla ja lukemalla alla olevaa listaa komenoista.

Super Star Trek

BASIC-kielinen pelitiedosto on peräi-

sin amerikkalaisilta tuttaviltani, joiden kanssa olemme yhdessä editoineet, korjanneet ja muokanneet sitä niin, että se toimii uudenaikaisessa PC-BASICissa nimeltä QB64. Kirjoitin tästä BASIC-variantista Skrollin numerossa 2015.4.

Ohjelma on niin vanha, että varsinaisen ohjelmakoodin on vanhaa niin sanottua rivinumero-BASICia. Ohjelmatedoston alussa on kommentteja, joista käy ilmi muun muassa mitä korjauksia ja muutoksia koodaajatt ovat tehneet Super Star Trek -ohjelmaan. Ohjelman lukemista hieman haittaa se, että alkuperäinen ohjelmoija ei ole käyttänyt itsedokumentoivia symbolien nimiä.

Pelitiedoston nimi on SUPER-STARTREK.BAS; sen voi ladata osoitteesta skrolli.fi/numerot. Koko listaus on tuhansia rivejä, joten jatkossa olen poiminut esiteltäväksi vain osia ohjelmasta.

QB64-järjestelmä

Paras näkemäni täysin ilmainen BASIC-järjestelmä, QB64, on ladattavissa osoitteesta www.qb64.org. Toisin kuin monet BASICit, QB64 ei ole tulkki vaan kääntäjä: aina käynnistettäessä ohjelmaa BASIC-koodi käännetään konekielelle. BASIC-ohjelman voi myös kääntää konekielelle ja tallentaa levyille – käännetty tiedosto jää käännökseen jälkeen samaan hakemistoon, missä .BAS-lähdetiedosto on ollut. Linuxissa ladattu paketti puretaan hakemistoon qb64, jonka jälkeen QB64-BASIC ladataan, käännetään ja asennetaan ajamalla komentotiedosto `setup_lnx.sh`. Windowsissa tai Macissa sovelletaan samaa toimintaa. Linuxin tapauksessa tiedostoa `setup_lnx.sh` pitää kuitenkin ensiksi muokata käytössä olevan jakelun mukaan: esimerkiksi Ubuntu Linuxissa tiedostoon kirjoitetaan `DOWNLOAD=1` ja `DISTRO=ubuntu`.

Itse QB64 käynnistetään komentotiedostolla `run_qb64.sh`.

Peliohjelman rakenne

Riveillä 19–27 on DATA-lauseissa ohjelman käyttämät merkkijonot, kuten käyttäjän komennot, sekä galaksin osien eli kvadranttien nimet. Niiden alle olen lisännyt ohjelmaan rivin 30

RANDOMIZE TIME, joka alustaa BASICin satunnaislukugeneraattorin PC:n reaaliaikakellosta. Tällöin saadaan joka kerralla uudenlainen pelikokemus.

Ohjelman pääsilmutka

Peli toimii koko pelin ajan silmukassa, joka alkaa ohjelman rivistä 70 ja jossa aina palataan riville 70. Rivillä 75 käyttäjä syöttää komennon. Riveillä 110–130 käyttäjän syöttämää merkkijonoa verrataan ohjelman komentoihin. Jos komento löytyy, rivillä 120 hypätään kahteen laskettuun GOTO (computed GOTO) -lauseeseen, joista lähtien suoritetaan käyttäjän antama komento. Nämä lasketut GOTO-lauseet ovat rivit 150 ja 160.

Lasketuista GOTO-lauseista siirrytään moniin eri ohjelmakohtiin, joista jokainen ohjelmakohta aina suorittaa yhden komennon.

Näissä ohjelmakohdissa hypätään aliohjelmiin GOSUB-lauseilla, ja nämä aliohjelmat suorittavat varsinaiset komennot. Kaikissa kohdissa, joissa aliohjelmista on palattu takaisin RETURN-lauseella, on lopussa lause GOTO 70, jossa palataan yllä mainittuun pelisilmukan alkuun – siis riville 70.

Ohjelman logiikka on siis yksinkertainen. Ainoa monimutkaisuus on siinä, että komentoja on monta, ja GOSUB-aliohjelmat, jotka suorittavat komennot, ovat kohtalaisen monimutkaisia. Näillä eväillä BASIC-kielilaitoinen lukija saa kuitenkin melko helposti selvää ohjelman koodista.

Galaksin tallettaminen taulukkoon

Galaksin kvadrantit eli osat on talletettu matriisiin `s2(8,8)`. Tässä BASIC-kielessä ei ollut olemassa tietuetietotyyppejä, joten jokaisen galaksin kvadrantin sisältämät objektit on talletettu yhteen kokonaislukuun, joka on matriisin `s2(*,*)` alkion arvona.

Tässä:

- Jos matriisin `s2(*,*)` alkio on negatiivinen, kvadrantissa on avaruustukikohta, eikä siellä ole vielä käyty;
- Jos matriisin alkio on nolla, galaksin kvadrantti on täysin tyhjä;
- Jos matriisin alkio on suurempi kuin 1000, kvadrantissa ei ole vielä käyty, eikä tiedetä, mitä siellä voisi olla;
- Jos matriisin alkio on esimerkiksi 315, kvadrantissa on 3 klingonia, yksi avaruustukikohta ja 5 tähteä. Asiaan liittyy vielä muutakin, jonka jätän lukijalle kotitehtäväksi.

Pelin komennot

Seuraavassa on listattu kaikki pelin komennot sekä niiden kohdat ohjelmassa.

ABANDON

Tämä käyttäjän käsky on GOSUB-rutiini, joka alkaa ohjelman riviltä 200.

ABANDON-komennolla miehistö jättää aluksen.

CHART

Tämä käsky on GOSUB-rutiini, joka alkaa ohjelman riviltä 225.

```
70 IF a2 <> 0 THEN 900
75 j4 = 0: t1 = 0: PRINT: INPUT "COMMAND "; a$: IF LEN(a$) > 1 THEN 110
80 PRINT "USE AT LEAST 2 LETTERS, PLEASE.": GOTO 75
110 FOR i = 1 TO 20
    120 IF a$ = LEFT$(c$(i), LEN(a$)) THEN 150
130 NEXT
135 INPUT "ILLEGAL COMMAND - DO YOU NEED A LIST "; b$
136 IF LEFT$(b$, 1) = "N" THEN 70
140 PRINT: FOR i = 1 TO 20 STEP 4
    141 PRINT c$(i); TAB(12); c$(i + 1); TAB(22); c$(i + 2); TAB(32); c$(i + 3)
142 NEXT: PRINT: GOTO 70
144 REM
146 REM THE COMPUTED GOTO'S, TO JUMP TO THE COMMAND TO EXECUTE:
148 REM
150 ON i GOTO 200, 225, 250, 275, 290, 300, 325, 350, 375, 400
160 ON i - 10 GOTO 425, 450, 475, 500, 525, 540, 550, 575, 600, 625
170 PRINT "ERROR AT 170 - SHOULD NOT BE HERE"
180 END
```

Listaus 2. Pääsilmutka. Rivit 70–180.

```
19 DATA "S.R. SENSORS", "L.R. SENSORS", "PHASERS", "PHOTON TUBES", "LIFE SUPPORT"
20 DATA "WARP ENGINES", "IMPULSE ENGINES", "SHIELDS", "SUBSPACE RADIO"
21 DATA "SHUTTLE CRAFT", "COMPUTER", "TRANSFER PANEL", "ABANDON", "CHART", "COMPUTER"
22 DATA "DAMAGES", "DESTRUCT", "DOCK", "IDLE", "IMPULSE", "LRSCAN", "NAVIGATE", "PHASERS", "QUIT"
23 DATA "SHIELDS", "SOS", "SRSCAN", "STATUS", "TORPEDO", "TRANSFER", "VISUAL", "WARP", "SHORT"
24 DATA "MEDIUM", "LONG", "BEGINNER", "NOVICE", "SENIOR", "EXPERT", "COURSE", "WCOST", "ICOST"
25 DATA "PEFFECT", "SCORE", "END", "ANTARES", "SIRIUS", "RIGEL", "MERAK", "PROCYON", "CAPELLA"
26 DATA "VEGA", "DENEK", "CANOPUS", "ALDEBARAN", "ALTAIR", "REGULUS", "BELLATRIX", "ARCTURUS"
27 DATA "POLLUX", "SPICA", 10.5, 12, 1.5, 9, 0, 3, 7.5, 6, 4.5
```

Listaus 1. Rivit 19–27.

CHART piirtää galaksista kartan. Kartta sisältää ne galaksin osat, jotka on jo nähty ja joiden sisältämät oliot tunnetaan.

COMPUTER

Tämä alirutiini alkaa riviltä 250. Se kutsuu Enterprisen tietokonejärjestelmää.

COMPUTER-komennolla on joukko alikomentoja, joilla lasketaan erilaisia hyödyllisiä asioita.

DAMAGES

Tämä GOSUB-aliohjelma alkaa riviltä 275. DAMAGES-komento listaa Enterprisen vauriot, jotka ovat syntyneet taistelussa.

DESTRUCT

Aliohjelma alkaa riviltä 291. DESTRUCT-komento tuhoaa oman aluksen, siis Enterprisen. (Miksi kukaan käyttäisi sitä?)

DOCK

Tämä rutiini alkaa riviltä 300. DOCK-komento edellyttää, että Enterprise sijaitsee aivan tukikohdan vieressä. Komennolla telakoidutaan tukikohtaan.

IDLE

Aliohjelma alkaa riviltä 325. Kuten englannin kielen verbi ”to idle” sanoo, tällä komennolla kulutetaan aikaa tekemättä mitään. Sinä aikana klingonit saattavat tosin hyökätä!

IMPULSE

Tämä GOSUB-rutiini alkaa riviltä 350. IMPULSE-komennolla lennetään ”impulssiajolla” eli rakettimoottoreilla.

LRSCAN

Rutiini alkaa riviltä 375. LRSCAN-komento kutsuu Long Range Scannereita, joilla näkee oman ja muutaman ympäröivän kvadrantin sisältämät oliot.

NAVIGATE

Aliohjelma alkaa riviltä 400. Tällä komennolla lennetään Enterprisen poimuajolla (warp drive). Komento kysyy kurssin (kellonaikasuunta kuten sotilaskäytössä) ja lennettävän etäisyyden kvadrantteina (saa olla desimaaliluku).

PHASERS

Rutiini alkaa riviltä 425. PHASERS-komennolla ammutaan vihollisia vaiheilla, jotka ovat toinen Enterprisen aseista. Vaiheisten käyttö vaatii tietyn määrän energiaa.

QUIT

Tämä rutiini alkaa riviltä 450. QUIT-komento lopettaa pelin kesken.

SHIELDS

Aliohjelma alkaa rivillä 475. SHIELDS-komennolla nostetaan tai lasketaan Enterprisen sädeasesuojakilvet, mihin kuluu tietty määrä energiaa.

SOS

Aliohjelma alkaa riviltä 501. SOS on hätäkutsu.

SRSCAN

Tämä alkaa riviltä 525. SRSCAN-komennolla kutsutaan Short Range Scannereita, joilla näkee oman kvadrantin sisällä olevat oliot.

STATUS

Aliohjelma alkaa rivillä 540. Komento näyttää Enterprisen tilan.

TORPEDOS

Tämä GOSUB-aliohjelma alkaa riviltä 550. Komennolla laukaistaan fotonitorpedot. Ne ovat tehokkaita aseita, mutta Enterprise voi kantaa niitä vain 10 kappaletta kerrallaan.

TRANSFER

Aliohjelma alkaa riviltä 575. Komennolla siirretään lisää energiaa sädesuojakilpiin. Tämä voi olla tarpeen kesken taistelun.

VISUAL

GOSUB-rutiini alkaa riviltä 600. Komennolla katsellaan omaa kvadranttia kiikareilla, jolloin näkyy tietty sektori katselusuuntaan. Tämä voi olla tarpeen, jos taistelussa Short Range Scannerit ovat vahingoittuneet.

WARP

Tämä alirutiini alkaa rivillä 625. Tällä komennolla asetaan Enterprisen poimujon (warp drive) nopeus (poimutekijä, warp factor). Se ei voi olla alle 1. 🚀

8.-10.9.2017 Tampere-talo
TRACON



Traconin peliperjantai:

8.9.2017

Tampere-talo

Chris Huelsbeck -konsertti

Kuule maineikkaan säveltäjän pelimusiikkia kolmen pianon sovituksina, mm. The Great Giana Sisters, Turrican... sekä Huelsbeck itse kertomassa työstään!

LIPUT: www.tampere-talo.fi

Ehkä olin siinä kymmenvuotias, kun tietotekniset härvelit herättivät huomioni. Luultavasti havahduin samoihin aikoihin pelihallipeleihin, ensimmäisiin taskuvideopeleihin, pelikonsoleihin ja kotimikroihin. Olin ollut ennenkin peli- ja vekotinhöpsö. Legot ja pienoismallit olivat oleellinen osa kasvuaani, sillä tykkäsin suunnitella, askarrella ja rakennella juttuja. Tietysti rakastin myös lautapelejä. Opettelin pelaamaan shakkia omin päin kirjaston kirjoista, joita saatoin lukea kassin, pari viikossa. Varhaisimpia pelimuistoja on upea ravipeli, jossa hevoset kiisivät raviradalla nopanheittojen myötä. Keksin hepoille nimiä ja tarinoita, etenkin aivan yksin pelatessani. Moninpeliä yksin, jep, tykkäsin introvertteillä jo silloin.

Jotenkin tämä tietotekniikan ja videopelien aalto oli kuin juuri minulle tarkoitettu. Minä, pelit ja tietokoneet olimme tarkoitettuja yhteen.

Ja siksi olen onnekas. Sain olla läsnä silloin, kun kaikki tämä saapui ja yleisty. Toki minua ennen oli vanhoja, viisaita miehiä ja naisia. 1970-luvun puolella oli pioneereja, jotka rakensivat koneensa rakennussarjoista ja ohjelmoivat itse niiden softat. Minun vallankumoukseni oli valmiimpi.

Kauppojen hyllyt täyttyivät valmiista kotimikroista ja niiden ohjelmista. Vähemmän yllättäen pelit olivat niitä tärkeimpiä. Tietokoneita myyviä putiikkeja oli jopa meidänkin pienessä kotikaupungissamme – ja paljon. Piuhat vain paikoilleen ja siinä meillä oli välittömästi suvun nuori tietokoneguru, jonka pelaamista sukulaiset katselivat kunnioittavasti.

”Kaikkea se osaakin”, tädit taivastelivat.

Kotimikro vai sähköurut?

Ensimmäisten kotimikrojen kohdeyleisö oli oikeastaan aika epämääräistä. Niitä kun myytiin mitä kummallisimmissa kaupoissa. Yleisimmin koneita kaupattiin kodinkoneliikkeissä, siinä missä televisioita, stereota ja jääkaappejakin. Myöhemmin koneita oli myös musiikkiliikkeissä. Osta Yamaha, oli se sitten mikro tai syntsa.

Myyjien asiantuntemus rajoittui usein siihen, mitä prosyyrissä luki, mutta kauppa kävi. Ostajat tiesivät ko-



En koe olevani vanha tai edes keski-ikäinen, vain onnekas. Nykyajon neilla on kivaa, kun digitaalinen maailma muuttuu heidän ympärillään ja asioita voidaan tehdä hiipaten ja näpläten. Mutta silloin kun minä jonneilin, maailma muuttui nopeammin ja draamatisemmin. Kaikin tavoin, ei vain digitaalisesti.

Teksti: Jukka O. Kauppinen Kuvat: Susanna Rantanen/Manu Pärssinen, Reima Mäkinen, Petri Tuomola, Triosoft

neista usein enemmän, ainakin kun asialla oli palavasilmäinen juniori, joka oli vertaillut esitteiden speksejä ja harkinnut äärimmäisen tosissaan minkä koneen leiriin hän asettuu. Joskus näet oli sekini aika, jolloin Commodore 64 ei ollut vielä kukkulan kuningas, vaan koneeksi saattoi valikoitua lähes mikä tahansa aikakauden suosikkimikro. Tällöin ratkaisu saattoi oikeasti pohjautua koneiden teknisiin ominaisuuksiin, toisinaan myös lisälaittevalikoimaan ja niiden hintaan. Olisikohan joskus tarvetta ihan printterillekin? Ei tuommoista levyasemaa nyt ainakaan tarvita, se on niin kalliskin.

Toisinaan myyjä onnistui puhumaan lompakonhaltijan ympäri, ja kotiin valikoituikin jokin aivan väärä kone. Voi niitä, jotka halusivat Commandoren mutta saivatkin Oricin.

Nykyisenkaltaisia vertailuja koneiden välillä ei tehty eikä pystyttykään tekemään. Esitteet olivat raamattuja, ja niihin piti luottaa. Ehkä se Amstradin peli on nykyisin katsottuna tosi

nykivä ja kökkö, mutta kyllä se oli ihmeellinen kaupan esittelykoneella pyöriessään.

Puodit olivat myös tärkeitä kokoon-tumispaikkoja. Niissä tulisilmäiset intoilijat tuijottivat ihmeellisiä koneita ja juttelivat. Uusia laitteita tuntui ilmestyvän yhtenä, ja jokaiselle oli omat pelinsä. Joskus kauppoihin tuli uusia pelejä, ehkä niitä sai kokeillakin. Jos kauppa laittoi ikkunaansa tietokoneen, niin uteliaita riitti. Meitä vakiokasvoja oli paljon, sillä piti hankkia tietokonekaupat kiertää läpi viikoittain ihan siltä varalta, että siellä olisi jotain uutta. Emme me yleensä mitään ostaneet, vaan pelasimme sitäkin innokkaammin. Myyjät olivat ymmärtäväisiä eivätkä ajaneet poikia pois. Kiitos siitä teille näin jälkikäteen.

Kaikkeen ei tosiaan ollut varaa, yleensä ei mihinkään, etenkin omaan koneeseen. Niillä onnekailla, joilla oli oma tietokone, riitti vieraita. Näpertäminen oli lähes yhteisöllistä. Luimme ohjekirjoja, naputimme listauksia,

opettelimme ja tietenkin pelasimme. Hauskaa oli.

Yhteisöllisyyttä ennen somea

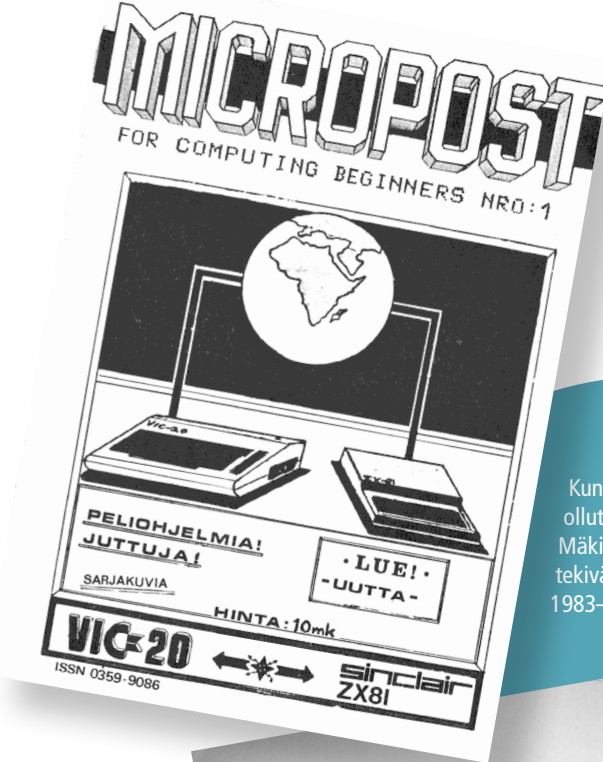
Koulujen tietokonekerhoilla oli tähän aikaan todella suuri merkitys. Niitä ei kaiketi pyöritetty kovinkaan järjestelmällisesti tai ylhäältä ohjattuna, mutta jos koulussa sattui olemaan tietotekniikasta innostunut opettaja, niin kerhoihin riitti osallistujia.

Vaikka touhu oli ainakin meillä vakavaa, niin silti se kiinnosti. Pelejä ei muistaakseni paljoa pelattu, vaan ainakin me opettelimme enemmänkin Basic-ohjelmointikieltä ja koneiden käyttöä. En minä edes Basiciä koskaan kunnolla oppinut, mutta muut kyllä. Kerhotoiminta oli kyllä kovasti kivaa, saihan siellä näpertää tietokoneita! Se oli tärkeintä, omaa kun ei vielä ollut. Oman kerhoni vetäjällä oli monta tietokonetta ja shakkitietokone, jota sain joskus lainata. Se kävi muuten niin kuumana, että muuntaja piti laittaa talvella roikkumaan johdosta ikkunan ulkopuolelle, ettei se ylikuumentunut. Opella oli myös suomenkielinen kummitustekstipeli, jota olen siitä lähtien yrittänyt löytää.

Kerhossa tapasi paikallisia bittivaihtajia, joista monilla oli tietysti jo oma kone! Joidenkin luona käytiin katsomassa ihmekoneita ja pelaamassa. Eräältä vierailulta mieleen jäivät vahvasti Commodoren International Soccer (1983), Hitchhiker's Guide to the Galaxy (1984) ja lentoyhtiöstrategiapeli High Flyer (1983), joista jälkimmäinen jätti pysyvän jäljen meikäläisen kehittyvään pelimakuun. Lentokoneita, bisnestä, managerointia ja strategiaa, kaikki ne maittavat edelleenkin.

Niin tietotekniikka saapui keskuuteemme, kodinkoneliikkeiden ja musakauppojen kautta. Tiedonsaanti oli yhä hyvin hataraa, sillä eihän Suomessa ollut vielä edes yhtään tietokonelehteä. Englanninkielisiä lehtiä ei kaupoissa ollut, eikä niitä varmaan olisi osattu lukeakaan. Jotenkin sitä vain silti juttuja kuultiin. Pelien ja koneiden mukana tulleet esitteet luettiin tarkasti läpi.

Yhä useampi sai oman koneen, pelejä vaihdeltiin ja tietoutta levitettiin. Hiljalleen opimme tekemään muuta kuin lataamaan kasetilta pelejä. Se on oma tarinansa se. 🎮



Kun tietokonelehtiä ei vielä ollut, niitä tehtiin itse. Reima Mäkinen ja Petri Tuomola tekivät Micropostia vuosina 1983–1985.

Ergonomiasta ja koneenkäytön mukavudesta ei tähän aikaan vielä välitetty. Tärkeintä oli, että sai naputtaa. Kuva: Reima Mäkinen.



Tältä se varmaan näytti, kun asuimme tietokonekaupoissa haaveillen omasta koneesta ja uutuuspeleihin tutustuen. Kuva: Trisoft.

Jukka O. Kauppinen alias Grendel/Byterapers on pitkän linjan toimittaja, joka kirjoitti ensimmäisiä tekstejään mekaanisella kirjoituskoneella. JOKin ensimmäinen peliarvostelu julkaistiin 1986, mistä lähtien hän on ahkeroinut tauotta kirjoittaen juttuja muun muassa videopeleistä, tietotekniikasta, viihteestä, ilmailusta ja burleskista. JOKistoriaa-sarjassa pureskellaan niin kotimikroilun kuin peli- ja digijournalismin maailmaa grennulasien läpi nähtynä. jukka@skrolli.fi



Kotiteatterin istuimet täyttyivät osallistujista.



Nasu "Bass Cadet" Viljanmaan musiikkiesitys lumosi katsojat.

Kiitos Skrollin avustajat!

Skrolli Party 2017

Skrolli Party 2017 järjestettiin Espoon Tuomarilassa 6.5.2017 kiitoksena lehden parissa puuhaaville parhaille ihmisille.

Teksti: Valhe Kouneli Kuvat: Mette Erikkilä

Aivan ensimmäiseksi: anteeksi, jos kutsu jäi saamatta. Teimme parhaamme.

Skrolli Party järjestettiin Skrollin avustajille – kirjoittajille, kuvittajille, ständipäivystäjille, roudareille ja muille puuhavoimille – kiitoksena arvokkaasta työstä ja yrityksenä kertoa, kuinka arvostamme teitä hurjasti. Tilaaajat ja lukijat olivat myös tervetulleita, mutta pyysimme osallistumisoikeudesta vastineeksi osanottoa kilpailuun digitaalisella teoksella.

Skrolli Partyjä juhlittiin perinteisten demopartyjen muodossa, eli ohjelmaan kuului oluen ja saunomisen lisäksi demoshown, tracker-musiikkia, eri digitaalisen taiteen muodot sekoittava kompo (kilpailu) ja kiinnostavia, tietoteknisesti orientoituneita ihmisiä. Skrollin pitkäaikainen avustaja **Mikko "tdb" Rasa** tarjosi ystävällisesti tapahtuman käyttöön paritaloasuntonsa, jonka kellarikerrokseen rakennettu kotiteatteri tarjosi hyvät puitteet demojen katsomiseksi ja musiikkiesityksille. Yläkerran valoisassa olohuoneessa oli mukava tutustua vieraisiin ja saada kasvot monelle pitkään vain nimenä Skrollin sivuilla ja irc-kanavalla pyörineelle henkilölle. Vietimme iltaa paikan päälle tilatun pizzan voimalla – valitettavasti sää ei houkuttellut grillaamaan ulkona, vaikka siihen oli mahdollisuus.

Erikoisohjelmaan kuului **Nasu**

"Bass Cadet" Viljanmaan musiikkiesitys thereminillä ja syntetisaattoreilla. Myös **Kimmo "dB" Rinta-Pollari** esitteli tracker-sävellyksiään.

Tapahtuma keräsi noin kaksikymmentä osallistujaa, joista suurin osa oli Skrollin avustajia. Toimituksen jäseniä oli paikalla kuusi. Suunnitelmissa on järjestää vastaava tapahtuma uudelleen vuoden sisään. Jos et päässyt osallistumaan tai kuullut tapahtumasta ajoissa, mutta haluaisit osallistua seuraavalla kerralla, laita viesti osoitteeseen party@skrolli.fi niin otamme yhteyttä, kun partyja taas juhlietaan!

Lisää kuvia Skrolli Partyiltä löydät osoitteesta party.skrolli.fi/kuvat2017.



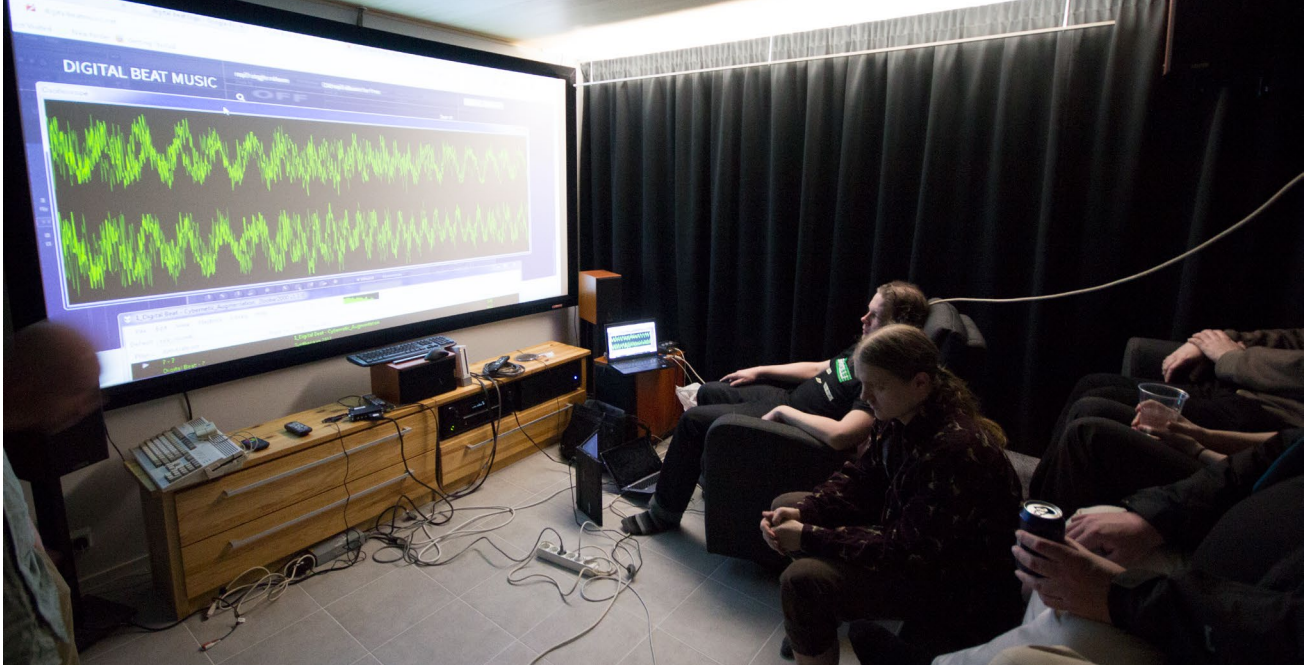
Pääjärjestäjä Mikko "tdb" Rasa.



Kimmo Rinta-Pollari esitteli musiikkiaan.



Olohuoneessa keskusteltiin tietokonekulttuurista ja rentouduttiin.



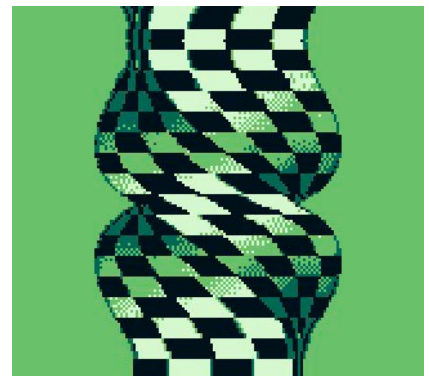
Jättimäinen valkokangas oli omiaan demojen katseluun. Tässä kuunnellaan musiikkia.

Kuten demopartyillä yleensä, myös Skrolli Partyillä osa kilpailutöistä aloitettiin tai saatettiin valmiiksi vasta paikan päällä. Kilpailuun osallistuneita töitä oli yhteensä yhdeksän, joista neljä oli demoja, neljä kuvia ja yksi musiikkiteos. Skrollin avustajat ja lukijoiden taitojen laaja-alaisuus tuli hyvin edustetuksi.

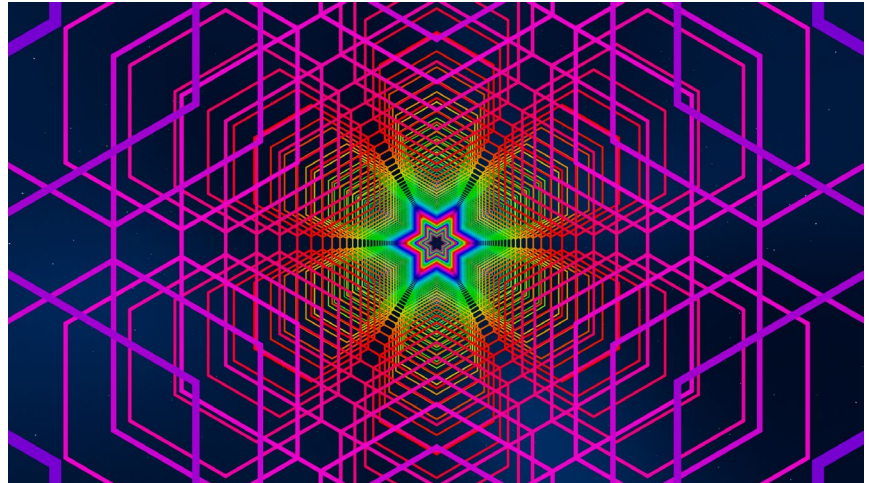
Partyen osallistajat pisteyttivät työt, minkä perusteella voittajaksi valikoitui **Deekun** ja **Imre Peemotin** Geometrify-moottorilla tehty **Skrollightenment**-demo. Demossa voi-

daan nähdä suuri Buddha-patsas Skrolli-lehti edessään, minkä jälkeen katsojia elähdytetään numerosta 2016.2 tutuilla geometrisilla kuvioilla ja astraalisella äänimaailmalla. Deeku sai voitosta palkinnoksi MIST-laitteen, joka osaa muuntautua eri retrokoneiksi. Skrollia tervehdittiin myös **naaviksen** Game Boylle tekemässä demossa **Roboto**, joka sijoittui kolmanneksi.

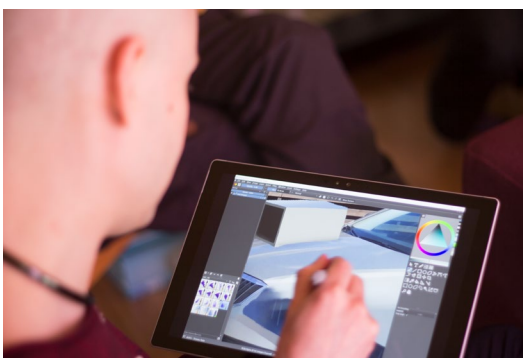
Kolme parasta voi ladata osoitteesta skrolli.fi/numerot. 🐛



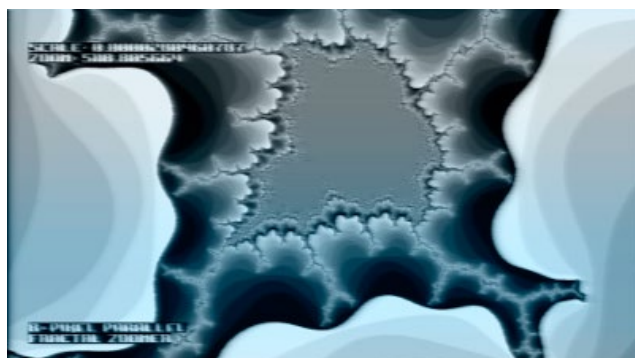
Roboto-demo.



Kompon Skrollightenment-demolla voittanut deeku istui teoksensa näyttämisen ajan meditatiivisessa risti-istunnassa mallintamansa Buddha-patsaan tapaan.



Zagupi loi kompotyönsä paikan päällä.



Toiseksi sijoittui Hackersin demo X220.

REAKTOR

HOW

ALSO
IN

TURKU

Apply now: turku@reaktor.com

Reaktor

